LONDON
METROPOLITAN
UNIVERSITY

CS6P05ES Project

FUTURE-TECH PROJECT

Name:	: MERINNAGE DON RUMAL THARINDA NAYANATHA
ID Number:	: COL 00155819
Student ID:	: E027616
Date:	: 20.07.2023
First Supervisor:	: Dr. Ranatunga

Declaration

Module: CS6P05ES **Deadline: 21.07.2023**
Module Leader: Dr. Ranatunga **Student ID:** COL 00155819

PLAGIARISM
You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed below. Please sign below to say that you have read and understand these extracts:

(Signature:) ____________________ Date: 20.072023

This header sheet should be attached to the work you submit. No work will be accepted without it.

Extracts from University *Regulations* on Cheating, Plagiarism and Collusion
Section 2.3: "The following broad types of offence can be identified and are provided as indicative examples…

(i) Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.

(ii) Falsifying data in experimental results.

(iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.

(iv) Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.

(v) Collusion to present joint work as the work solely of one individual.

(vi) Plagiarism, where the work or ideas of another are presented as the candidate's own.

(vii) Other conduct calculated to secure an advantage on assessment.

(viii) Assisting in any of the above.

Some notes on what this means for students:

1. Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.

2. Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication,

2
e.g. "e = mc (Einstein 1905)". A *references* section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

Dedication

This document is dedicated to DR Ranathunge, whose dedication, passion, and vision have been instrumental in the development of the shopping cart and stock management web application for Future Tech. Your unwavering commitment to excellence and your leadership have guided the project towards success. Your tireless efforts and invaluable insights have shaped the direction and implementation of this solution.

We express our deepest appreciation for your guidance and support throughout the project. Your unwavering commitment to innovation and your drive to meet the evolving needs of Future Tech and its customers have been truly inspiring. This document stands as a testament to your vision and the collaborative effort of the entire team.

Thank you, M P S Wijesinghe, for your relentless pursuit of excellence and for being a driving force behind the success of this project. Your dedication and expertise have been crucial in shaping the shopping cart and stock management web application, enabling Future Tech to thrive in the competitive retail industry.

Acknowledgments

I take this opportunity to express my heartfelt gratitude to all the individuals who have extended their guidance and support, contributing to the successful completion of this project. Firstly, I am immensely thankful to ESOFT Metro Campus for providing us access to textbooks and the ELMS platform, which played a crucial role in collecting relevant data and deepening our understanding of the subject.

A special appreciation goes to our esteemed lecturer, Dr. Ranathunge, whose invaluable subject knowledge, guidance, and encouragement have been instrumental in turning this project into a reality. Additionally, I would like to extend my thanks to our previous lecturers, whose teachings laid the foundation for our understanding of computers and the IT industry.

Finally, I want to extend my gratitude to our family members and friends for their unwavering support and encouragement throughout this journey. Their belief in us has been a source of motivation and inspiration.

Abstract

The retail industry is undergoing a significant transformation with the rise of e-commerce and online shopping. To stay competitive and cater to evolving customer demands, Future Tech, a leading computer and computer parts selling company, recognized the need to adapt and leverage technology. This project aims to develop a shopping cart and stock management web application for Future Tech, providing an intuitive and seamless online shopping experience while streamlining stock management processes.

The implementation of the project followed the waterfall methodology, encompassing planning, analysis, design, implementation, testing, and maintenance phases. The tools and technologies employed included HTML, CSS, PHP, NetBeans, SQL, phpMyAdmin, and XAMPP. The system architecture adopted a three-tier model, consisting of the presentation layer, application layer, and data layer.

The web application's functionalities include user registration and authentication, product catalog and search, shopping cart functionality, secure payment gateway integration, order processing and tracking, inventory management, customer relationship management, reporting and analytics, mobile responsiveness, and security and data privacy measures.

Throughout the development process, an evaluation was conducted to assess the solution's feasibility, functionality, user experience, integration capabilities, security, performance, scalability, and maintenance. The evaluation confirmed the successful fulfillment of requirements and objectives, highlighting the solution's effectiveness in addressing Future Tech's challenges and improving business processes.

In conclusion, the developed shopping cart and stock management web application have empowered Future Tech to adapt to the changing retail landscape, enhance customer satisfaction, expand its customer base, and improve operational efficiency. By leveraging the implemented solution and continuously refining it based on user feedback and market trends, Future Tech can position itself as a trusted and customer-centric online retailer in the computer parts industry.

Table of Contents

Table of Contents

Declaration 2

Dedication.. 3

Acknowledgments 4

Abstract.. 5

Table of Contents.. 6

Chapter 1 -Introduction 9

1.1 Aim 9

1.2 Objectives 9

1.3 Motivation 10

1.4 Method.. 11

1.5 Overview 12

Chapter 2 -Background and Problem Statement 13

2.1 Introduction 13

2.2 Literature Review 13

2.3 Analyses Requirements 14

2.4 Similar System.. 16

2.5 Problem Statement.. 17

Chapter 3 -Project Management 18

3.1 Introduction 18

3.2 Approach 19

3.3 Initial Project Plan 20

3.4 Problems and Changes to the Plan.. 25

3.5 Final Project Record 25

Chapter 4 -Feasibility Study 26

4.1 Introduction 26

4.2 Schedule Feasibility. 26

4.3 Financial Feasibility. 26

4.4 Legal and Regulatory Feasibility 26

4.5 Technical Feasibility. 26

4.6 Operational Feasibility. 26

Chapter 5 -Design 28

5.1 Introduction 28

5.2 Requirements Definition 28

5.3 Functional and Non-Functional Requirements 30

5.4 System Design 32

5.3 Hardware and software requirements 98

5.4 Evaluating of solutions 99

Chapter 6 -Implementation 101

6.1 Introduction 101

6.2 Implementation Methodology 101

6.3 System Architecture 103

6.4 Tools & Technologies used 104

6.5 Database Implementation 105

6.6 User Interface Design 109

6.7 Backend Code Implementation 115

Chapter 7 -Testing and Verification 119

7.1 Introduction 119

7.2 Aim of Testing 119

7.3 Test Plan 120

7.4 Validation Test Plan 122

7.5 Validation Test Case....124

Chapter 8 -Evaluation and Conclusion....139

8.1 Summarized Evaluation....139

8.2 Conclusion....140

9. References....141

10. Appendices....143

Chapter 1 -Introduction

1.1 Aim

The aim of the Future Tech shopping cart and stock management web application project is to leverage technology to transform Future Tech into a leading online retailer of computer parts and accessories. By developing a user-friendly and efficient web application, the aim is to enhance customer satisfaction, expand market reach, and optimize stock management processes.

1.2 Objectives

1. Develop a User-Friendly Shopping Cart:
    - Design and implement an intuitive and user-friendly shopping cart module that allows customers to easily browse, select, and purchase computer parts and accessories online.
    - Enable customers to create accounts, manage their profiles, and track their orders through the shopping cart.

2. Streamline Stock Management:
    - Develop a comprehensive stock management module that enables Future Tech to accurately track inventory levels, monitor product availability, and prevent stockouts.
    - Implement automated alerts and notifications for low stock levels, enabling timely reordering and replenishment.

3. Enhance Customer Experience:
    - Create a personalized experience for customers by implementing features such as product recommendations, wishlists, and ratings/reviews.
    - Provide a seamless and secure checkout process, integrating with popular payment gateways to facilitate smooth transactions.

4. Improve Operational Efficiency:
    - Develop modules for employee management, enabling efficient tracking of employee roles, responsibilities, and performance.
    - Implement a supplier management module to streamline communication and collaboration with suppliers, ensuring a reliable supply chain.

5. Gather Actionable Insights:
    - Design and implement a reporting and analytics module to capture and analyze customer data, enabling Future Tech to gain insights into customer preferences, purchase patterns, and trends.
    - Utilize data-driven insights to optimize marketing strategies, enhance product offerings, and improve customer targeting.

6. Ensure Scalability and Security:
    - Build a scalable web application architecture that can handle increasing user traffic and growing product catalog.
    - Implement robust security measures, including encryption, user authentication, and data privacy, to protect customer information and ensure a secure shopping environment.

By achieving these objectives, the project aims to position Future Tech as a reputable online retailer, offering a seamless shopping experience, a wide range of products, efficient stock management, and exceptional customer service.

1.3 Motivation

The motivation behind the Future Tech shopping cart and stock management web application project stems from several key factors:

1. Evolving Customer Preferences: In today's digital age, customers increasingly prefer the convenience of online shopping. By developing an online platform, Future Tech can cater to changing customer preferences and provide a seamless shopping experience that aligns with modern expectations.

2. Market Expansion and Reach: The physical store of Future Tech limits its geographical reach and potential customer base. By establishing an online presence, Future Tech can overcome this limitation and expand its market reach, tapping into a broader customer demographic and potentially increasing sales.

3. Operational Efficiency: Manual stock management processes often result in inefficiencies, inaccuracies, and stockouts. By implementing a robust stock management system within the web application, Future Tech can streamline its inventory control, reduce errors, and improve overall operational efficiency.

4. Competitive Advantage: The introduction of a shopping cart and stock management web application gives Future Tech a competitive edge in the computer parts retail industry. It positions the company as an innovative and customer-centric organization, capable of providing a modern shopping experience that differentiates it from competitors.

5. Data-Driven Insights: The web application enables Future Tech to gather valuable customer data, such as purchase history, preferences, and behavior. By analyzing this data, Future Tech can gain actionable insights to tailor marketing strategies, improve product offerings, and enhance customer engagement, ultimately driving sales growth.

6. Adaptation to Technological Advancements: As technology continues to advance, businesses must embrace digital transformation to remain relevant and competitive. The Future Tech project reflects the company's commitment to leveraging technology to meet the demands of the evolving marketplace and stay ahead of industry trends.

Overall, the motivation behind the project lies in Future Tech's desire to meet customer expectations, expand its market reach, improve operational efficiency, gain a competitive advantage, and adapt to the digital era. By embracing the web application, Future Tech aims to enhance the customer experience, drive business growth, and establish itself as a leader in the computer parts retail industry.

1.4 Method

Throughout the development process, it is crucial to adhere to software development best practices, maintain effective communication with project advisors or mentors, and ensure compliance with relevant ethical and legal considerations. The following method will be employed to ensure the successful implementation of the project objectives:

1. Requirement Gathering:

- Conduct research and analysis to understand the needs and expectations of Future Tech's target customers.
- Gather requirements for the shopping cart, stock management, customer experience, operational efficiency, data analytics, scalability, and security aspects of the web application.
- Consult with Future Tech's management and stakeholders to align project goals and objectives.

2. System Design:

- Create a comprehensive system design that includes the architecture, database schema, user interface, and module specifications.
- Consider factors such as usability, scalability, security, and integration with external systems (e.g., payment gateways, supplier communication).
- -Seek feedback and validation from project advisors or mentors to ensure the design meets industry standards and best practices.

3. Development and Implementation:

- Utilize appropriate programming languages, frameworks, and technologies to develop the web application.
- Follow an iterative development approach, regularly testing and refining the functionality and user interface.
- Implement the shopping cart module, stock management module, customer experience features, operational efficiency modules, data analytics module, and security measures according to the project requirements.

4. Testing and Quality Assurance:

- Conduct thorough testing to identify and resolve any functional or performance issues.
- Perform unit testing, integration testing, and system testing to ensure the web application functions as intended.
- Apply quality assurance techniques to verify that the web application meets predefined standards and specifications.

5. Deployment and Hosting:

- Deploy the web application to a suitable hosting environment, ensuring it is accessible to users via the internet.
- Configure the necessary server infrastructure, database management, and security measures to support the application's operation.

6. User Training and Documentation:

- Provide training sessions or documentation to Future Tech's employees on how to use and manage the web application.
- Prepare comprehensive user manuals and documentation to guide users through the features and functionalities of the web application.

7. Evaluation and Feedback:

- Seek feedback from Future Tech's management, employees, and potential users to assess the usability, effectiveness, and performance of the web application.
- Incorporate feedback and suggestions into future iterations or updates of the web application to enhance its functionality and user experience.

By following this method, the Future Tech shopping cart and stock management web application project aims to deliver a high-quality, user-friendly, and efficient solution that meets the defined objectives and provides value to Future Tech and its customers.

1.5 Overview

The Future Tech shopping cart and stock management web application project aims to transform Future Tech into a leading online retailer of computer parts and accessories. By leveraging technology, the project aims to enhance customer satisfaction, expand market reach, and optimize stock management processes.

The project's objectives encompass various key aspects, including the development of a user-friendly shopping cart, streamlined stock management, enhanced customer experience, improved operational efficiency, gathering actionable insights, and ensuring scalability and security.
To achieve these objectives, a systematic method will be followed, starting with requirement gathering to understand customer needs and align project goals. A comprehensive system design will be created, considering factors such as architecture, database schema, and user interface. The development and implementation phase will involve building the shopping cart module, stock management module, customer experience features, operational efficiency modules, data analytics module, and security measures. Rigorous testing and quality assurance will be conducted to ensure the web application's functionality and performance.

The deployed web application will be hosted on a suitable infrastructure, and Future Tech's employees will receive training and documentation on its usage and management. Continuous evaluation and feedback will be sought to improve the application's usability and effectiveness, incorporating suggestions into future updates.

The motivation behind the project stems from evolving customer preferences, the need for market expansion and reach, operational efficiency improvements, gaining a competitive advantage, accessing data-driven insights, and adapting to technological advancements.
By successfully executing the project, Future Tech aims to establish itself as a reputable online retailer, offering a seamless shopping experience, efficient stock management, and exceptional customer service. The web application will position Future Tech as an innovative and customer-centric organization, capable of meeting the demands of the digital era and staying ahead in the competitive computer parts retail industry.

Chapter 2 -Background and Problem Statement

2.1 Introduction

The retail industry is undergoing a significant transformation with the advent of e-commerce and online shopping. Future Tech, a leading computer and computer parts selling company based in Colombo, recognizes the need to adapt to this changing landscape to stay competitive and cater to the evolving customer demands. With five years of experience in the industry, Future Tech aims to leverage technology to enhance its sales, expand its customer base, and improve overall operational efficiency.

2.2 Literature Review

The literature review explores existing research and studies related to online shopping platforms and stock management systems in the retail industry. It provides insights into the best practices, features, and benefits offered by these systems, which can guide the development of Future Tech's shopping cart and stock management web application.

1. Online Shopping Platforms:

Numerous studies have highlighted the advantages of online shopping platforms in the retail industry. Chen and Barnes (2007) emphasize that online shopping platforms offer convenience, accessibility, and a wider product selection, resulting in increased customer satisfaction and loyalty. Moreover, Liu and Li (2014) found that online shopping platforms positively influence customer perceptions of product quality and brand reputation.

2. User Experience and Interface Design:

The user experience (UX) and interface design of online shopping platforms play a crucial role in attracting and retaining customers. Research by Wu, Liao, and Chen (2016) reveals that a well-designed user interface, intuitive navigation, and personalized recommendations contribute to higher user satisfaction and increased purchase intention. Additionally, Kim and Lennon (2013) emphasize the importance of a visually appealing and user-friendly interface in enhancing the overall shopping experience and customer engagement.

3. Secure Payment and Trust:

Security is a significant concern in online transactions. A study by Fang, Chiu, and Wang (2011) emphasizes that integrating secure payment gateways and encryption mechanisms into online shopping platforms enhances customer trust and reduces perceived risks. Trust is a critical factor influencing customer loyalty and repeat purchases (Chiu et al., 2014), indicating the importance of secure payment integration in Future Tech's web application.

4. Inventory Management Systems:

Effective stock management systems are crucial for retailers to maintain optimal inventory levels, prevent stockouts, and ensure timely order fulfillment. A study by Dong, Xu, and Song (2015) highlights the importance of real-time inventory updates, accurate demand forecasting, and automated replenishment processes in minimizing stock-related issues and maximizing operational efficiency. Implementing these features in Future Tech's web application will contribute to improved inventory management.

5. Customer Relationship Management (CRM):

CRM systems enable retailers to capture and analyze customer data, facilitate personalized interactions, and enhance customer satisfaction. Research by Rigby, Reichheld, and Schefter (2002) suggests that CRM implementation positively impacts customer loyalty and profitability. Additionally, Alalwan et al. (2017) highlight the significance of customer data analytics in identifying customer preferences, tailoring marketing strategies, and delivering personalized experiences.

6. Mobile Responsiveness and Multi-Channel Integration:

With the increasing use of mobile devices for online shopping, mobile responsiveness is crucial for a seamless user experience. Research by Barnes and Corbitt (2003) emphasizes the importance of mobile optimization and responsive design in attracting and retaining mobile shoppers. Furthermore, integrating the online platform with other sales channels, such as physical stores or marketplaces, can provide a unified customer experience and enhance brand visibility (Brynjolfsson and Smith, 2000).

By considering the insights from these studies and implementing best practices observed in successful online shopping platforms, Future Tech can develop a comprehensive web application that addresses the challenges outlined in the problem statement. The literature review provides a strong foundation for understanding the key features and benefits that should be incorporated into Future Tech's shopping cart and stock management system to enhance customer satisfaction, improve operational efficiency, and position itself as a competitive player in the computer parts retail industry.

2.3 Analyses Requirements

1. User Registration and Authentication:
   - The system should allow customers to register and create user accounts.
   - User authentication mechanisms, such as username/password or social media login, should be implemented to ensure secure access to the system.

2. Product Catalog and Search:
   - The application should provide a comprehensive product catalog with detailed information about each item, including pricing, specifications, and availability.
   - A search functionality should be implemented to allow customers to easily find desired products based on various criteria, such as keywords, categories, and brands.

3. Shopping Cart Functionality:
   - The system should enable customers to add products to a virtual shopping cart, view the cart contents, and modify quantities or remove items.
   - Customers should be able to proceed to checkout and initiate the payment process from their shopping cart.

4. Secure Payment Gateway Integration:

- The application should integrate with secure payment gateways to facilitate secure online transactions.
- Multiple payment options, such as credit/debit cards, online banking, and digital wallets, should be supported to accommodate customer preferences.

5. Order Processing and Tracking:

- The system should automate the order processing workflow, including order confirmation, invoice generation, and order status updates.
- Customers should be able to track the status of their orders and receive notifications about important updates, such as order shipment or delivery delays.

6. Inventory Management:

- The application should provide real-time inventory management capabilities to track product stock levels and prevent overselling.
- Automatic stock updates should occur when orders are placed, ensuring accurate availability information for customers.

7. Customer Relationship Management (CRM):

- The system should capture and store customer data, such as contact information, purchase history, and preferences.
- CRM functionality should enable personalized customer interactions, targeted marketing campaigns, and customer support activities.

8. Reporting and Analytics:

- The application should generate reports and provide analytics features to track sales performance, inventory turnover, and customer behavior.
- Data visualization tools can be utilized to present actionable insights that help drive business decisions and improve operational efficiency.

9. Mobile Responsiveness:

- The web application should be designed with responsive layouts and optimized for mobile devices, ensuring a seamless shopping experience across different screen sizes and resolutions.

10. Security and Data Privacy:

- The system should incorporate appropriate security measures, including data encryption, secure transmission protocols, and secure user authentication.
- Compliance with data protection regulations, such as GDPR or relevant local privacy laws, should be ensured to protect customer data and privacy.

These requirements provide a foundation for developing the shopping cart and stock management web application. It is essential to further analyze and refine these requirements based on the specific needs and goals of Future Tech.

2.4 Similar System

In the retail industry, there are several existing systems that provide online shopping and stock management capabilities, catering to the needs of businesses in the same domain. These systems have been developed by various companies with the aim of facilitating efficient online sales and streamlining stock management processes. Notable examples include Laptop.lk, a leading online retailer in Sri Lanka specializing in laptops and computer accessories. Laptop.lk offers a robust system that enables customers to browse a wide range of products, compare specifications, and make purchases with secure online payment options. Their system also features a user-friendly interface that enhances the overall shopping experience, ensuring that customers can easily find the products they need.

Winsoft.lk is another prominent player in the online retail space, focusing on software and gaming products. Their system provides a user-friendly interface, a streamlined checkout process, and reliable delivery services for its customers. Winsoft.lk's extensive catalog includes a wide selection of software titles, gaming accessories, and hardware components. With their commitment to customer satisfaction and prompt order fulfillment, Winsoft.lk has established a strong presence in the market.

SENSE.lk is recognized for offering a comprehensive online shopping experience for consumer electronics and home appliances. Their system includes detailed product descriptions, customer reviews, and personalized recommendations, providing valuable information for customers to make informed purchasing decisions. With a user-centric approach, SENSE.lk ensures a seamless and convenient shopping experience, complemented by secure payment options and responsive customer support.

In addition to Laptop.lk, Winsoft.lk, and SENSE.lk, bigdeals.lk and daraz.lk are also prominent players in the Sri Lankan online retail landscape. bigdeals.lk provides an extensive range of products across various categories, offering customers a diverse selection, competitive pricing, and reliable delivery services. daraz.lk, on the other hand, stands out as one of the largest online marketplaces in Sri Lanka. Their system caters to a wide range of product categories, featuring a comprehensive seller management platform for businesses to efficiently list and manage their inventory. daraz.lk prioritizes secure payment options, order tracking, and nationwide delivery services to ensure a trustworthy and convenient shopping experience.

These systems, including Laptop.lk, Winsoft.lk, SENSE.lk, bigdeals.lk, and daraz.lk, have been successfully implemented in Sri Lanka, contributing to the growth of online retail and enhancing the customer shopping experience. Future Tech aims to draw inspiration from these successful systems while incorporating specific customizations and enhancements to meet its unique requirements and provide an exceptional user experience tailored to the computer and computer parts selling industry. By leveraging the strengths and best practices of these existing systems, Future Tech aims to create a platform that offers an extensive product selection, streamlined ordering process, secure transactions, and reliable delivery services, positioning itself as a trusted and customer-centric online retailer in the market.

2.5 Problem Statement

The current business operations at Future Tech face several challenges that hinder its ability to meet customer expectations and achieve optimal efficiency. These challenges include:

1. Limited Reach: Future Tech's physical store restricts its customer base to those who can visit the showroom, limiting its potential for growth and expansion.

2. Manual Stock Management: Future Tech relies on manual processes for stock management, leading to inaccuracies, stockouts, and difficulties in tracking inventory levels. This results in missed sales opportunities and customer dissatisfaction.

3. Inefficient Sales Process: The absence of a comprehensive online shopping platform makes the sales process cumbersome and time-consuming for both customers and employees. This leads to lower customer satisfaction and potential revenue loss.

4. Lack of Data Insights: Future Tech lacks a centralized system for capturing and analyzing customer data, limiting its ability to understand customer preferences, improve marketing strategies, and personalize the shopping experience.

To address these challenges and unlock new growth opportunities, Future Tech seeks to develop a shopping cart and stock management web application. This web application will provide an intuitive and convenient online shopping experience for customers while enabling Future Tech to streamline its stock management processes, gain valuable insights, and improve overall operational efficiency.

By developing this web application, Future Tech aims to elevate its business to the next level, enhance customer satisfaction, expand its customer base, and position itself as a leading player in the computer parts retail industry.

Chapter 3 -Project Management

3.1 Introduction

The Future Tech shopping cart and stock management web application is an ambitious project aimed at revolutionizing the operations of Future Tech, a prominent computer and computer parts selling company based in Colombo. With five years of experience in the industry, Future Tech aims to enhance its sales and expand its customer base through the implementation of a modern online shopping platform.

The primary objective of this project is to develop a comprehensive web application that seamlessly integrates various modules to streamline employee management, customer interactions, product catalog management, stock control, supplier management, courier services, and an intuitive shopping cart system. By leveraging the power of the internet, Future Tech intends to leverage its online presence to attract new customers and provide an enhanced shopping experience for existing ones.

The web application will feature distinct modules that cater to the needs of different stakeholders. The employee module will facilitate efficient management of employee information, roles, and responsibilities. The customer module will provide a user-friendly interface for customers to browse and purchase products, manage their accounts, and track their orders. The product module will enable Future Tech to showcase its extensive range of computer parts and accessories, complete with detailed descriptions and specifications. The stock module will ensure accurate inventory control, allowing Future Tech to manage product availability and restocking efficiently. The supplier module will facilitate seamless communication and collaboration with suppliers, ensuring a reliable supply chain. The courier module will enable smooth coordination of order deliveries, ensuring timely and efficient shipment of products. Finally, the shopping cart module will provide customers with a convenient and secure online shopping experience, allowing them to add products, make payments, and track their orders.

By implementing this web application, Future Tech aims to enhance its operational efficiency, improve customer satisfaction, and boost sales. Customers will have the flexibility to browse products, place orders, and make payments from the comfort of their homes. The comprehensive stock management system will help Future Tech monitor product availability, ensure timely restocking, and prevent stockouts. Moreover, the integration of a courier module will enable efficient order fulfillment and delivery, further enhancing the customer experience.

With a clear vision and a commitment to technological advancement, Future Tech is poised to embark on this transformative project. By embracing e-commerce and investing in a robust web application, Future Tech is ready to redefine the way it does business and solidify its position as a leading provider of computer parts and accessories in the region.

3.2 Approach

To successfully develop the Future Tech shopping cart and stock management web application, the project will follow a systematic and iterative approach. The approach will involve several key stages, including requirements gathering, design, development, testing, and deployment. Here is an outline of the approach:

1. Requirements Gathering:
    - Engage with stakeholders at Future Tech to understand their specific needs and goals for the web application.
    - Conduct interviews and workshops to gather detailed requirements for each module, such as employee management, customer interactions, product catalog, stock control, supplier management, courier services, and shopping cart functionality.
    - Analyze existing processes and workflows to identify pain points and areas for improvement.

2. Design:
    - Based on the gathered requirements, design the overall architecture of the web application, ensuring scalability, security, and usability.
    - Create wireframes and prototypes to visualize the user interfaces of each module, considering ease of use and intuitive navigation.
    - Design the database schema to effectively store and retrieve data related to employees, customers, products, stock, suppliers, and orders.

3. Development:
    - Implement the web application using appropriate technologies and frameworks, such as HTML, CSS, JavaScript, and a backend language like Python or PHP.
    - Develop separate modules for employee management, customer interactions, product catalog, stock control, supplier management, courier services, and shopping cart functionality.
    - Integrate the modules to ensure seamless data flow and functionality across the application.
    - Implement security measures, including user authentication and authorization, to protect sensitive data.

4. Testing:
    - Perform comprehensive testing of each module to identify and fix any bugs or issues.
    - Conduct functional testing to ensure all features and functionalities work as intended.
    - Perform performance testing to assess the web application's responsiveness and scalability under various load conditions.
    - Conduct usability testing to gather feedback from users and make necessary improvements to enhance the user experience.

5. Deployment:
    - Prepare the web application for deployment on a web server or cloud hosting platform.
    - Configure the necessary infrastructure, including servers, databases, and network settings.
    - Migrate data from existing systems, if applicable, to the newly developed web application.
    - Conduct thorough testing in the production environment to ensure all components are functioning correctly.

6. Post-Launch Support:

- Provide ongoing support and maintenance for the web application, addressing any issues or bugs that arise.
- Monitor the application's performance, security, and user feedback to make continuous improvements.
- Regularly update the application to incorporate new features, enhancements, and security patches as needed.

By following this structured approach, the Future Tech shopping cart and stock management web application will be developed systematically, ensuring that it meets the specific requirements of the company and provides an exceptional user experience.

3.3 Initial Project Plan

1. Project Initiation Phase:

- Define project objectives, scope, and deliverables.
- Identify project stakeholders and establish communication channels.
- Set up project management tools and infrastructure.
- Assign project roles and responsibilities.

2. Requirements Gathering Phase:

- Conduct meetings and workshops with Future Tech stakeholders to gather detailed requirements for each module.
- Document functional and non-functional requirements.
- Prioritize requirements based on their importance and feasibility.
- Validate requirements with stakeholders and obtain their sign-off.

3. Design Phase:

- Design the overall architecture of the web application, considering scalability, security, and usability.
- Create wireframes and prototypes to visualize the user interfaces of each module.
- Design the database schema to efficiently store and retrieve data.
- Define the technical stack and frameworks to be used.

4. Development Phase:

- Develop the web application modules iteratively, starting with core functionalities.
- Implement employee management, customer interactions, product catalog, stock control, supplier management, courier services, and shopping cart modules.
- Integrate the modules to ensure seamless data flow and functionality across the application.
- Implement security measures, including user authentication and authorization.

5. Testing Phase:

- Conduct unit testing for individual modules to ensure their correctness.
- Perform integration testing to validate the interaction between different modules.
- Conduct functional testing to ensure all features and functionalities work as expected.
- Perform performance testing to assess the application's responsiveness and scalability.
- Conduct usability testing to gather user feedback and make necessary improvements.

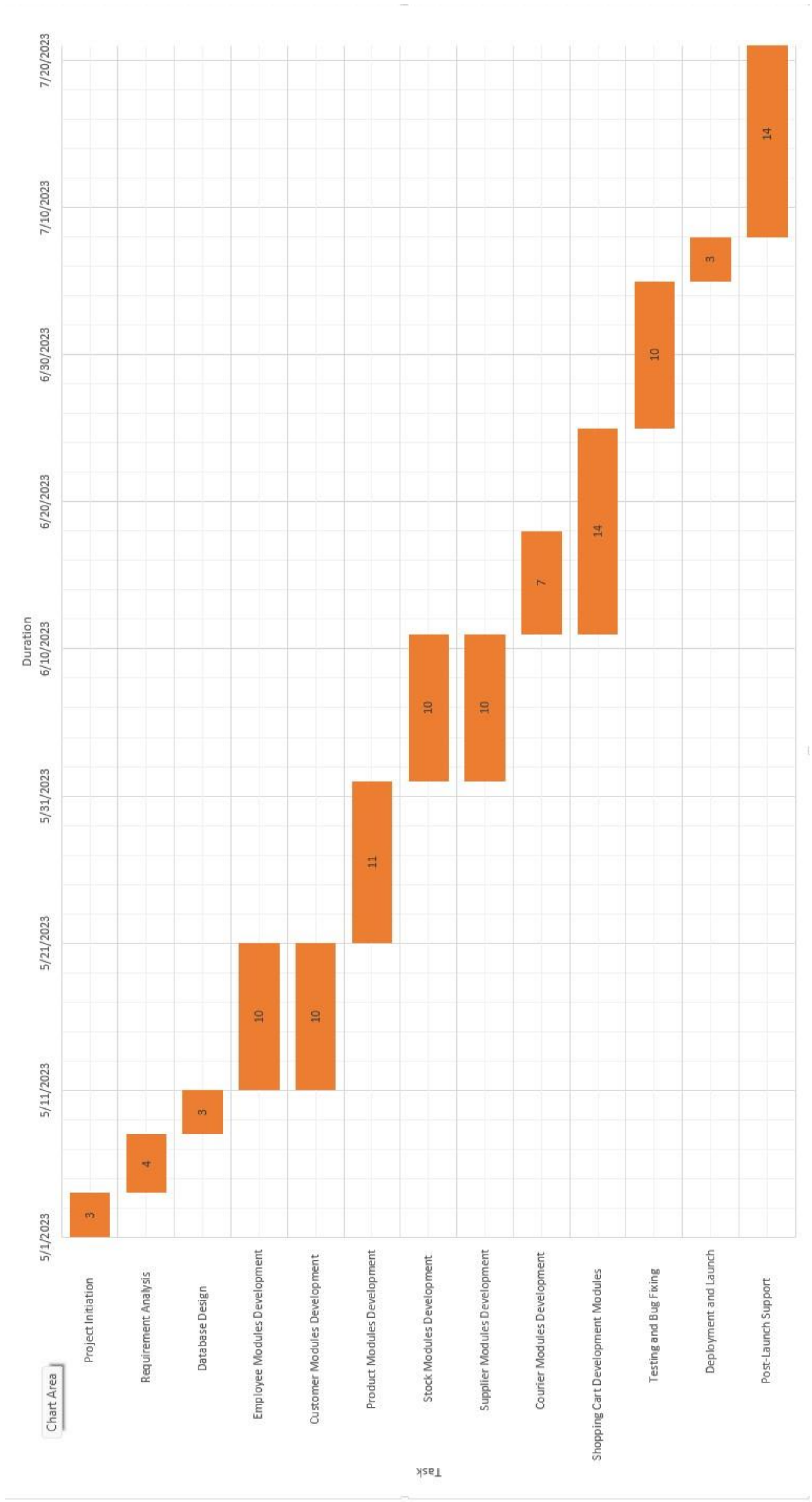
6. Deployment Phase:

- Prepare the web application for deployment on a web server or cloud hosting platform.
- Set up the necessary infrastructure, including servers, databases, and network settings.
- Migrate data from existing systems, if applicable.
- Conduct thorough testing in the production environment.
- Train Future Tech employees on using the web application.

7. Post-Launch Support Phase:

- Provide ongoing support and maintenance for the web application.
- Monitor the application's performance, security, and user feedback.
- Address any issues or bugs that arise.
- Regularly update the application to incorporate new features, enhancements, and security patches as needed.

3.3.1 Gantt Chart



3.3.2 Work Break Down Structure

1. Project Initiation
    - Define project objectives, scope, and deliverables
    - Identify stakeholders and establish communication channels
    - Set up project management tools and infrastructure

2. Requirements Gathering
    - Conduct meetings with Future Tech stakeholders to gather requirements
    - Document functional and non-functional requirements
    - Prioritize and validate requirements

3. Design
    - Design overall application architecture
    - Create wireframes and prototypes for each module
    - Design database schema

4. Development
    - Develop core functionalities of the web application
    - Implement employee management module
    - Implement customer module
    - Implement product module
    - Implement stock module
    - Implement supplier module
    - Implement courier module
    - Implement shopping cart module

5. Testing
    - Conduct unit testing for each module
    - Perform integration testing to ensure modules work together
    - Conduct functional testing to validate features and functionalities
    - Perform performance testing
    - Conduct usability testing

6. Deployment
    - Prepare web application for deployment
    - Set up necessary infrastructure and hosting
    - Migrate data, if applicable
    - Conduct production testing
    - Train Future Tech employees on using the web application

7. Post-Launch Support
    - Provide ongoing support and maintenance for the application
    - Monitor application performance and user feedback
    - Address any issues or bugs that arise
    - Continuously update and enhance the application based on feedback

3.3.3 Task Distribution & Tracking

1. Task Distribution:

- Review the work breakdown structure (WBS) and identify individual tasks within each phase.
- Prioritize the tasks based on their dependencies and criticality.
- Assign tasks to yourself, considering your skills, expertise, and availability.
- If a task requires significant effort or expertise, consider breaking it down into smaller subtasks to facilitate easier management and tracking.

2. Task Tracking:

- Set up a project management tool or system to track and manage your tasks. This can be a simple to-do list, a spreadsheet, or a specialized project management software.
- Break down each task into smaller, actionable steps or subtasks.
- Estimate the duration for each subtask based on your judgment and experience.
- Set deadlines or target completion dates for each subtask.
- Regularly update the status of tasks as you make progress.
- Use progress indicators such as "Not Started," "In Progress," and "Completed" to track the status of each task.
- Keep track of any dependencies between tasks to ensure that you tackle them in the correct order.
- Identify any potential roadblocks or bottlenecks that may affect task completion and plan accordingly.

3. Task Prioritization:

- Continuously assess the priority of tasks based on their urgency and impact on the project.
- Consider any dependencies between tasks and prioritize those that are critical for the completion of subsequent tasks.
- Review deadlines and adjust them if needed to ensure a realistic and achievable timeline.
- Communicate any changes or updates to the task priorities to stakeholders, if applicable.

4. Regular Progress Monitoring:

- Schedule regular checkpoints to review the progress of tasks and evaluate if adjustments are needed.
- Track your actual progress against the planned timeline and make adjustments as necessary.
- Identify any potential risks or issues that may arise and proactively address them to avoid delays or complications.
- Keep a record of completed tasks and milestones achieved to track your overall progress and accomplishments.

3.4 Problems and Changes to the Plan

During the course of the project, several problems and changes to the initial plan were encountered. This section highlights the key challenges and modifications that occurred.

Scope Expansion: As the project progressed, additional features and functionalities were identified and requested by stakeholders. This led to an expansion of the project scope, requiring adjustments to the project plan and timeline.

Technical Challenges: The implementation of certain functionalities, such as secure payment gateway integration and mobile responsiveness, posed technical challenges that required additional time and resources to overcome. These challenges resulted in modifications to the development approach and timeline.

Resource Constraints: Limited availability of skilled resources, such as developers and designers, posed challenges in meeting project deadlines. This necessitated resource allocation adjustments and re-prioritization of tasks.

Stakeholder Feedback: Feedback from stakeholders, including Future Tech management and end-users, resulted in changes to the user interface design, workflow, and functionality. These changes required iterative development and additional testing to ensure alignment with stakeholder expectations.

Technical Dependencies: The integration with external systems, such as payment gateways and shipping providers, introduced dependencies that required coordination and communication with third-party vendors. Delays or changes in the availability or functionality of these systems impacted the project plan.

3.5 Final Project Record

The final project record serves as a comprehensive documentation of the shopping cart and stock management web application development process. It includes all relevant project artifacts, such as project plans, requirement documents, design specifications, codebase, test cases, and user manuals.

The final project record represents the collaborative efforts of the development team, which includes you as the primary developer, alongside any other contributors or stakeholders involved in the project. It reflects the collective achievements, milestones, and outcomes of the project's lifecycle.

The record captures the project's progression, challenges encountered, decisions made, and solutions implemented by the development team. It serves as a valuable resource for future reference, maintenance, and enhancements of the web application.

In the case of a single developer, the final project record still encompasses all the relevant artifacts created and maintained by you throughout the development process. It showcases your individual contributions, dedication, and expertise in implementing the shopping cart and stock management web application.

The final project record stands as a testament to your hard work, skills, and achievements as the primary developer, ensuring the availability of comprehensive information and documentation for the shopping cart and stock management web application.

Chapter 4 -Feasibility Study

4.1 Introduction

This feasibility report assesses the viability of developing a shopping cart and stock management system for the future-tech. The objective is to streamline the purchasing process, improve inventory management, and enhance overall customer satisfaction. The report evaluates the technical, financial, legal and, operational and schedule feasibility aspects of the project.
The purpose of this feasibility study is to determine the potential success of the proposed shopping cart and stock management system. It aims to identify potential challenges, risks, and benefits associated with the development and implementation of the system.

4.2 Schedule Feasibility.

A realistic project timeline has been developed, accounting for all phases of development, testing, and implementation. Potential risks and challenges that may cause delays have been identified, and contingency plans are in place to mitigate their impact.

4.3 Financial Feasibility.

The estimated development and implementation costs for the shopping cart and stock management system have been evaluated. These include hardware, software, personnel, and other expenses. The projected revenue from increased sales and cost savings resulting from improved inventory management indicate a positive return on investment. The payback period is estimated to be within an acceptable timeframe, and the project is financially feasible.

4.4 Legal and Regulatory Feasibility

The proposed system will comply with all relevant legal and regulatory requirements, including data protection laws, payment processing regulations, and consumer rights. No significant legal or regulatory barriers have been identified that would hinder the development and implementation process.

4.5 Technical Feasibility.

As the developer of the system possesses the required technical expertise and capabilities to build the shopping cart and stock management system. The proposed system is compatible with the existing IT infrastructure and can be implemented using standard development technologies. No significant technical constraints or limitations have been identified.

4.6 Operational Feasibility.

The shopping cart and stock management system aligns with the existing business processes and workflows of the computer and computer parts seller. It will enhance operational efficiency, automate manual tasks, and improve order fulfillment. Adequate training and change management strategies will be implemented to facilitate a smooth transition and minimize disruptions during implementation.

Chapter 5 -Design

5.1 Introduction

In this chapter, we will delve into the design phase of developing the shopping cart and stock management web application for Future Tech. Designing a comprehensive and user-friendly online platform is crucial to meet the requirements identified in Chapter 2 and ensure a seamless shopping experience for customers. This chapter will outline the design principles, methodologies, and considerations that will guide the development process.

5.2 Requirements Definition

Based on the analyses conducted in Chapter 2, the following requirements have been identified for the shopping cart and stock management web application:

1. User Registration and Authentication:
   - The system should allow customers to register and create user accounts.
   - User authentication mechanisms, such as username/password or social media log in, should be implemented to ensure secure access to the system.

2. Product Catalog and Search:
   - The application should provide a comprehensive product catalog with detailed information about each item, including pricing, specifications, and availability.
   - A search functionality should be implemented to allow customers to easily find desired products based on various criteria, such as keywords, categories, and brands.

3. Shopping Cart Functionality:
   - The system should enable customers to add products to a virtual shopping cart, view the cart contents, and modify quantities or remove items.
   - Customers should be able to proceed to checkout and initiate the payment process from their shopping cart.

4. Secure Payment Gateway Integration:
   - The application should integrate with secure payment gateways to facilitate secure online transactions.
   - Multiple payment options, such as credit/debit cards, online banking, and digital wallets, should be supported to accommodate customer preferences.

5. Order Processing and Tracking:
   - The system should automate the order processing workflow, including order confirmation, invoice generation, and order status updates.
   - Customers should be able to track the status of their orders and receive notifications about important updates, such as order shipment or delivery delays.

6. Inventory Management:

- The application should provide real-time inventory management capabilities to track product stock levels and prevent overselling.
- Automatic stock updates should occur when orders are placed, ensuring accurate availability information for customers.

7. Customer Relationship Management (CRM):

- The system should capture and store customer data, such as contact information, purchase history, and preferences.
- CRM functionality should enable personalized customer interactions, targeted marketing campaigns, and customer support activities.

8. Reporting and Analytics:

- The application should generate reports and provide analytics features to track sales performance, inventory turnover, and customer behavior.
- Data visualization tools can be utilized to present actionable insights that help drive business decisions and improve operational efficiency.

9. Mobile Responsiveness:

- The web application should be designed with responsive layouts and optimized for mobile devices, ensuring a seamless shopping experience across different screen sizes and resolutions.

10. Security and Data Privacy:

- The system should incorporate appropriate security measures, including data encryption, secure transmission protocols, and secure user authentication.
- Compliance with data protection regulations, such as GDPR or relevant local privacy laws, should be ensured to protect customer data and privacy.

These requirements provide a clear roadmap for designing the shopping cart and stock management web application. By addressing each requirement and considering user experience, security, and scalability, Future Tech can develop a robust and user-centric platform that meets the needs and expectations of its customers.

5.3 Functional and Non-Functional Requirements

5.3.1 Functional Requirement

1. User Registration and Authentication:
The system should allow users to register accounts, log in, and authenticate their identity to access personalized features, view order history, and manage their shopping carts.

2. Product Catalog Management:
The system should provide functionality for managing a comprehensive product catalog, including adding, updating, and removing products. It should support categorization, product descriptions, images, and pricing information.

3. Shopping Cart Management:
The system should enable users to add products to their shopping carts, view and modify the contents of their carts, and proceed to the checkout process. It should handle multiple products, quantities, and variations.

4. Inventory Management:
The system should keep track of product inventory levels and update them in real-time as users add products to their carts or complete purchases. It should provide alerts when stock levels reach minimum thresholds and prevent users from purchasing out-of-stock items.

5. Product Search and Filtering:
The system should include search functionality to allow users to search for products based on specific criteria such as keywords, categories, brands, or price ranges. It should also support filtering options to refine search results.

6. Order Processing and Payment:
The system should facilitate order processing by allowing users to review their orders, select preferred payment methods, and complete secure transactions. It should integrate with payment gateways to handle payment processing.

7. Shipping and Delivery Management:
The system should support the selection of shipping options, calculation of shipping costs, and tracking of order shipments. It should integrate with shipping providers and provide real-time updates on delivery status.

8. Promotions and Discounts:
The system should enable the application of promotional offers, discounts, and coupon codes to eligible products or orders. It should calculate the discounted prices accurately during the checkout process.

9. Customer Reviews and Ratings:
The system should provide functionality for customers to leave reviews and ratings for products they have purchased. It should allow users to view and sort reviews to make informed purchasing decisions.

10. Reporting and Analytics:
The system should generate reports and analytics related to sales, inventory levels, customer behavior, and other relevant metrics. It should provide insights for business decision-making and performance evaluation.

5.3.2 Non-Functional Requirement

Security:

- The system provides username and password to prevent the system from unauthorized access.
- The staffs' password must be greater than eight characters.

Performance:

- The system response time for every instruction conducted by the user must not exceed more than a minimum of 10 seconds.
- The system should have high performance rate when executing user's input and should be able to provide response within a short time span usually 50 second for highly complicated task and 20 to 25 seconds for less complicated task.

Availability:

- The system should always be available for access at 24 hours, 7 days a week. Also in the occurrence of any major system malfunctioning, the system should be available in 1 to 2 working days, so that business process is not severely affected.

Usability:

- The system should have an intuitive and user-friendly interface that is easy to navigate, allowing customers to find products quickly and place orders without confusion. The interface should also be responsive and accessible across different devices and screen sizes.

Compatibility:

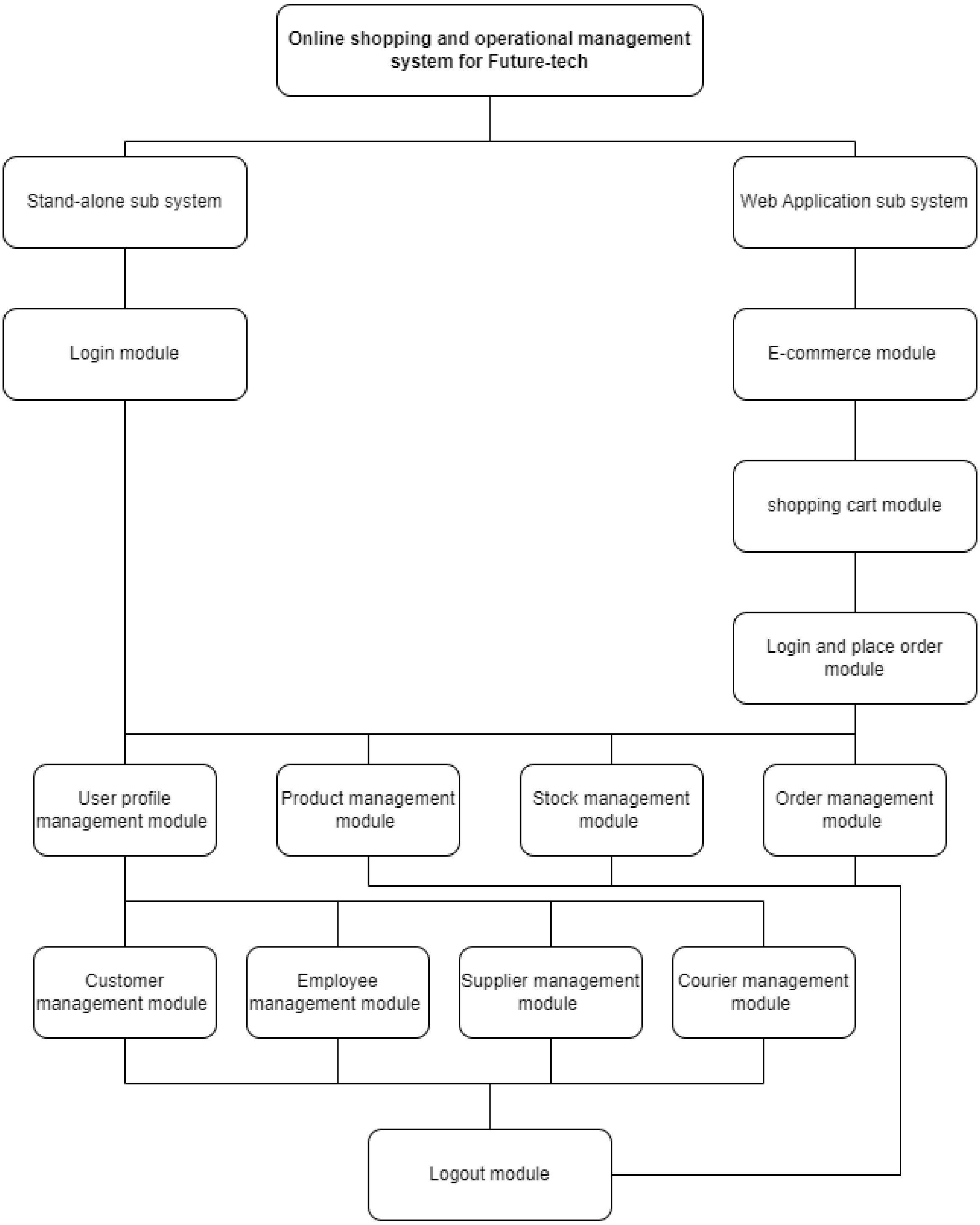
- The shopping cart system should be compatible with different browsers, operating systems, and devices (e.g., desktops, mobile devices). It should be responsive and adapt to various screen sizes and resolutions.

Maintainability:

- The system should be modular and well-structured, allowing for easy maintenance, updates, and enhancements. It should support efficient troubleshooting, bug fixing, and system upgrades without causing significant disruptions to the overall functionality.

5.4 System Design

Code and Module structure



Online shopping and operational management system for Future-tech
Stand-alone sub system
Web Application sub system
Login module
E-commerce module
shopping cart module
Login and place order module
User profile management module
Product management module
Stock management module
Order management module
Customer management module
Employee management module
Supplier management module
Courier management module
Logout module

The system design phase is a crucial step in the development of the shopping cart and stock management web application. It involves translating the requirements and architecture into a detailed design that defines the structure and components of the system. This section provides an overview of the key aspects of system design.

1. User Interface Design:

- Design the user interface (UI) to provide an intuitive and user-friendly experience.
- Create wireframes, prototypes, or mockups to visualize the layout, navigation, and interactions.
- Consider usability principles, such as consistent navigation, clear labeling, and responsive design for different devices.

2. Application Design:

- Identify the main components and modules of the application based on the requirements.
- Define the interactions and workflows between different components.
- Determine the data flow and communication protocols between the presentation layer, application layer, and data layer.
- Consider modular design principles to promote reusability and maintainability.

3. Database Design:

- Design the database schema based on the requirements and data model.
- Identify the tables, columns, relationships, and constraints.
- Optimize the database design for efficient data retrieval, storage, and manipulation.
- Consider database normalization techniques to eliminate redundancy and ensure data integrity.

4. Business Logic Design:

- Define the business rules and logic that govern the behavior of the application.
- Identify the functions, algorithms, and calculations required to implement the business logic.
- Determine how data will be processed, validated, and transformed within the application.

5. Integration Design:

- Plan the integration of external systems, services, or APIs required for payment processing, shipping, or other functionalities.
- Identify the necessary interfaces, protocols, and data formats for integration.
- Define the data exchange mechanisms and error handling strategies for seamless integration.

6. Security Design:

- Design the security measures to protect the application and user data.
- Implement authentication and authorization mechanisms to control access to the system.
- Apply encryption and hashing techniques to secure sensitive data.
- Consider security best practices, such as input validation, protection against SQL injection, and protection against cross-site scripting (XSS) attacks.

7. Error Handling and Logging Design:

- Define the error handling and logging mechanisms to capture and handle system errors or exceptions.
- Implement appropriate error messages and notifications for users.
- Set up logging mechanisms to record system events and facilitate troubleshooting and debugging.

8. Performance and Scalability Design:

- Design the system to handle expected user load and ensure optimal performance.
- Consider techniques such as caching, query optimization, and load balancing to enhance system performance.
- Plan for scalability by designing components to handle increased traffic or data volume.

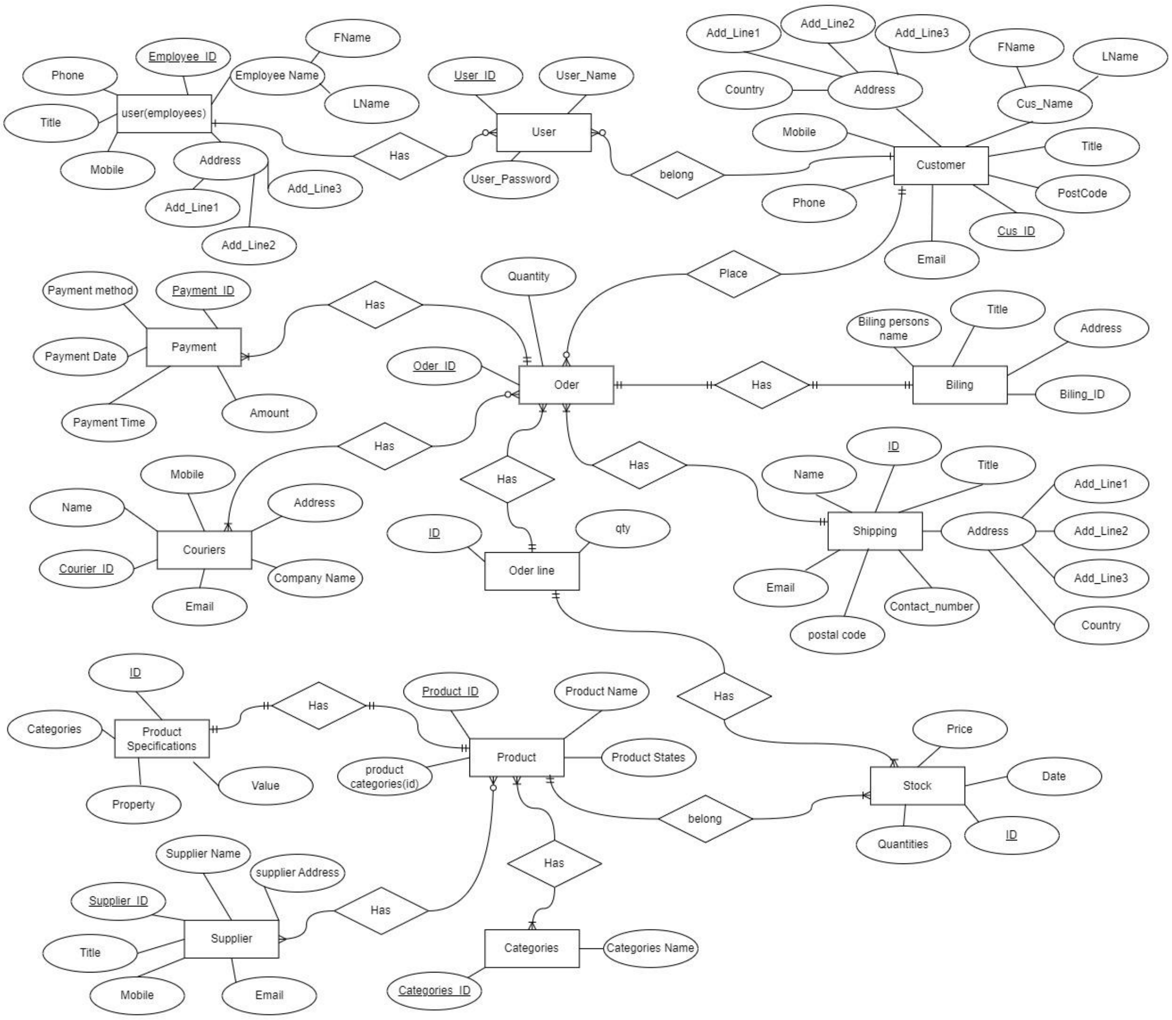
9. Documentation and Diagrams:

- Create documentation, including system architecture diagrams, entity-relationship diagrams, and sequence diagrams.
- Document the design decisions, assumptions, and rationale for future reference and knowledge sharing.

By addressing these aspects of system design, Future Tech can create a well-defined and robust design for the shopping cart and stock management web application. This design will serve as a blueprint for the implementation phase and guide the developer in building a functional, secure, and efficient system that meets the requirements and goals of Future Tech.

5.4.1 Entity Relationship Diagram

1.1 Redefined Entity Relational Ship Diagram of System



Relational Schema

- USER (user id, customer id*, employee id*, user name, user password)

- EMPLOYEES (employee id, employee first name, employee last name, title, address line1, address line 2, address line 3, mobile, phone)

- CUSTOMER (customer id, customer first name, customer last name, title, address line 1, address line 2, address line 3, country, postcode, email, phone, mobile)

- ORDER (order id, customer id*, couriers id*, shipping id*, order line id*, quantity)

- PAYMENT (payment id, order id* payment method, payment date, payment time, Amount)

- BILLING (billing id, order id*, billing person's name, billing person's title, address line 1, address line 2, address line 3)

- ORDER_LINE (order line id, quantity, shipping id*)

- COURIERS (couriers id, courier first name, courier last name, courier title, company name, address line1, address line 2 address line 3, phone, mobile)

- SHIPPING (shipping id, shipping first name, shipping last name, title, shipping address line 1, shipping address line 2, shipping address line 3, country, postal code, contact number, email)
- SHIPPING_CONTACT_NUMBER (contact id, shipping id* contact number, type)

- STOCK (stock id, order line id*, product id*, quantity, date, price)

- PRODUCT (product id, product name, product category, product states)

- PRODUCT_SUB_IMG (img_id, product_id*, img_file)

- PRODUCT_ SPECIFICATIONS (product spec id, product id*, category, property, value)

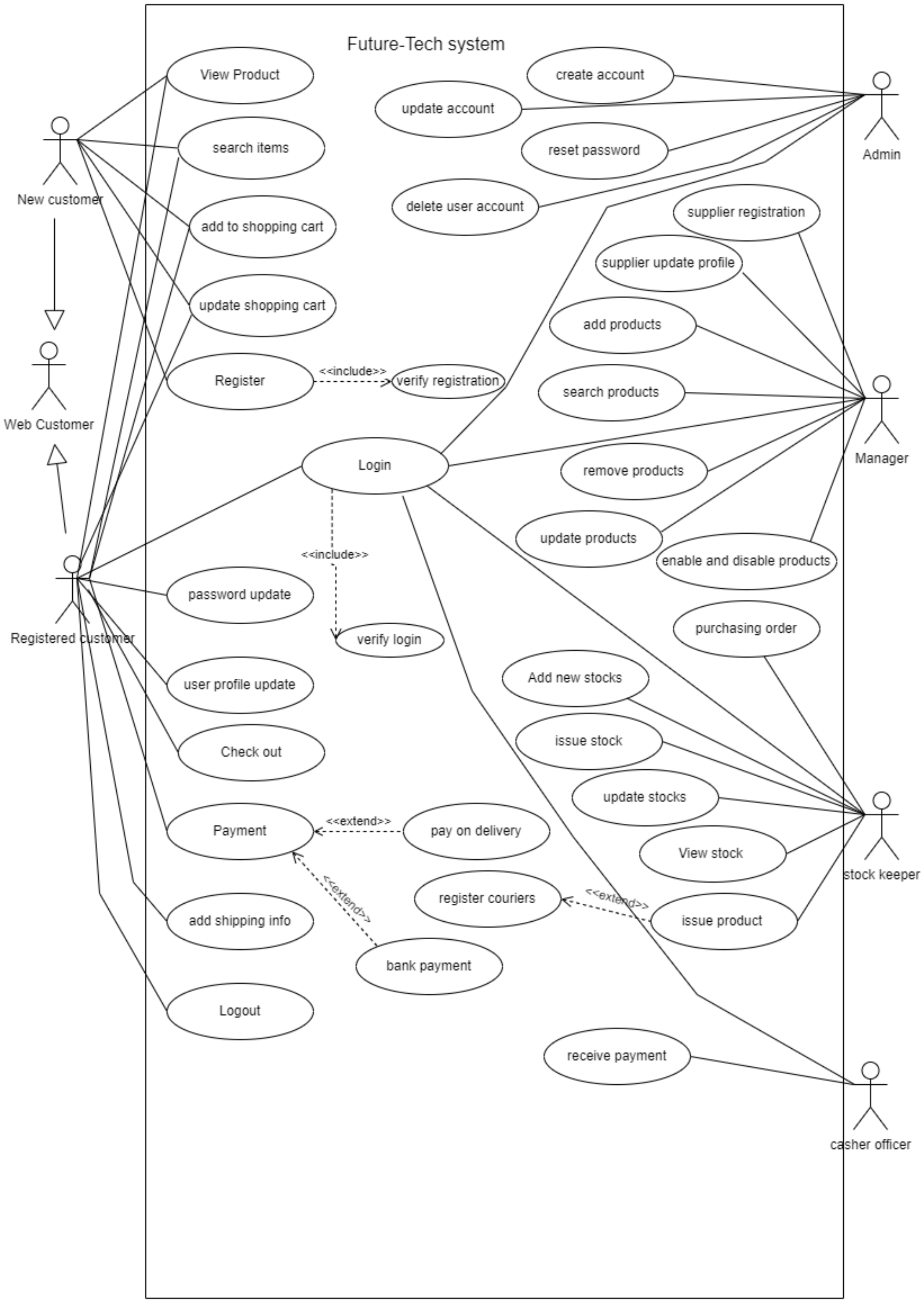
- CATEGORY (category id, category name)

- SUPPLIER (supplier id, supplier name, supplier address line 1, supplier address line 2, supplier address line 3, mobile, phone, email)

- SUPPLIER_PRODUCT (suppler_product_id, product id*, supplier id*, quantity, date, price)

- PRODUCT_CATEGORY (product_category_id, product id*, category id*)

1.2 Use case Diagram



Use-case Description

Use-case Login

Table 2.1: Use Case- Login

Use-case Number	UC-01	
Use-Case Name	Log in	
Priority	High	
Actor/s	Customer, Admin, Manager, stock keeper, sales officer	
Description	This use case describes how Customers and Staffs to login into the Future-Tech System.	
Precondition	Must create a User Account	
Post-condition	If the use case was successful, the actor is now logged into the system. If not, the system state is unchanged.	
Basic Course of Action	**User Action**	**System Response**
	1. The Actor clicks on the login button.	2. The system displays the login form to enter Username and Password.
	3. The Actor enters a Username and Password and clicks on a Login Button.	4. The system verifies that all the filled have been filled out and are valid. 5. The system successfully logged in the system. 6. Use case Exit
Alternate Course of Action	4.1 If all fields are not filled out and not matched to the username and password the system notifies the actor with a message “Invalid Username or Password” and then goes back or returns to step 3 of the basic course of Action to enter again.	

Use-case View Product

Table 2.2: Use Case - View Product

<table>
<tr><td>Use-Case Number</td><td colspan="2">UC-02</td></tr>
<tr><td>Use-Case Name</td><td colspan="2">View Product</td></tr>
<tr><td>Priority</td><td colspan="2">High</td></tr>
<tr><td>Actor</td><td colspan="2">Customer</td></tr>
<tr><td>Description</td><td colspan="2">This use case permits the customer to view the details and information of a specific product available on the Future-Tech website.</td></tr>
<tr><td>Precondition</td><td colspan="2">UC-01
The customer has navigated to the product listing or search page</td></tr>
<tr><td>Post-condition</td><td colspan="2">• The customer has reviewed the details and information of the selected product.
• The customer may have taken additional actions, such as adding the product to their shopping cart, wish list, or performing other actions related to the product.</td></tr>
<tr><td rowspan="2">Basic Course of Action</td><td><b>User Action</b></td><td><b>System Response</b></td></tr>
<tr><td>1. The customer enters a search query or browses through the product listing.
3. The customer selects a specific product from the list.
5. The customer reviews the product description, specifications, and any additional information provided by the system.
6. If applicable, the customer selects specific options or variants, such as size, color, or quantity, to customize the product.
8. The customer examines any available images or visual representations of the product, enabling them to evaluate its appearance or features.
9. If interested, the customer may add the product to their shopping cart or take other</td><td>2. The system retrieves a list of products matching the search query or available in the selected category.
4. The system displays the product information to the customer, including the product name, description, price, and any available options or variants.
7. The system updates the displayed information based on the selected options or variants.
11. Use case Exit</td></tr>
</table>

	relevant actions, such as adding it to a wish list or comparing it with other products. 10. The customer can choose to continue browsing for more products or perform other actions related to the viewed product, such as leaving a review or rating.	
Alternate Course of Action	8.1 If the customer does not find the desired product in the initial search or listing, they can refine their search criteria or navigate to different product categories. 8.2 If the selected product is out of stock or unavailable, the system notifies the customer and provides alternative options if available.	

Use-case Search items

Table 2.3: Use Case- Search items

Use-case Number	UC-03	
Use-Case Name	Search items	
Priority	High	
Actor/s	Customer	
Description	The customer wants to search for specific items on the Future-Tech website.	
Precondition	The customer is logged into the e-commerce website. The customer is on the homepage or a designated search page.	
Post-condition	• The customer has searched for specific items and reviewed the search results. • The customer may have taken additional actions, such as adding items to their shopping cart, wishlist, or performing other actions related to the searched items.	
Basic Course of Action	**User Action**	**System Response**
	1. The customer enters a search query into the search bar or selects predefined filters to narrow down their search. 5. The customer reviews the search results, scanning through the displayed items to find relevant ones. 6. If necessary, the customer can refine their search query or adjust the selected filters to further narrow down the results. 8. The customer selects a specific item from the search results to view its detailed information. 10. The customer can review the detailed information of the selected item to assess its suitability. 11. If interested, the customer may take additional actions, such as adding the item to their shopping cart, adding it to a wish list, or comparing it with other items. 12. The customer can choose to continue browsing or perform another search for different items.	2. The system validates the search query or selected filters. 3. The system retrieves a list of items matching the search query or selected filters. 4. The system displays the search results to the customer, including relevant items' names, images, prices, and any available options or variants. 7. The system updates the displayed search results based on the refined search query or adjusted filters. 9. The system presents detailed information about the selected item, including its name, description, price, available variants or options, and relevant images. 13. Use case Exit
Alternate Course of Action	2.1 If the search query does not yield any results, the system notifies the customer and suggests alternative search terms or filters. 2.2 If the selected filters yield no results, the system notifies the customer and suggests modifying or removing some of the filters.	

Use-case Add to shopping cart

Table 2.3: Use Case- Add to shopping cart

Use-case Number	UC-04	
Use-Case Name	Add to shopping cart	
Priority	High	
Actor/s	Customer	
Description	The customer wants to add a specific item to their shopping cart.	
Precondition	• UC-01 • UC-03	
Post-condition	• The customer has successfully added the selected product to their shopping cart. • The customer may have taken additional actions, such as adding more products to their shopping cart or proceeding to review and complete their purchase.	
Basic Course of Action	**User Action**	**System Response**
	1. The customer selects a specific item they wish to add to their shopping cart. 2. The customer selects the "Add to Cart" button associated with the item. 6. The customer may choose to continue browsing for more items or proceed to the checkout process.	3. The system verifies the availability of the item and validates the customer's request. 4. If the item is available and the desired quantity is feasible, the system adds the item to the customer's shopping cart. 5. The system updates the shopping cart, displaying the updated number of items and the total value. 7. Use case Exit
Alternate Course of Action	4.1 If the item is unavailable or the desired quantity exceeds the stock limits, the system notifies the customer of the unavailability and provides alternative options if applicable. 4.2 If the customer attempts to add more items to the cart than are available in stock, the system can either limit the quantity to the available stock or notify the customer of the stock limitation.	

Use-case Update shopping cart

Table 2.3: Use Case- Update shopping cart

Use-case Number	UC-05	
Use-Case Name	update shopping cart	
Priority	High	
Actor/s	Customer	
Description	The customer wants to update the quantity of an item in their shopping cart.	
Precondition	• UC-01 • UC-04	
Post-condition	• The customer has successfully removed the selected item from their shopping cart. • The shopping cart reflects the updated list of items and recalculates the total value if necessary. • The customer may choose to continue browsing, add more items to the cart, or proceed to the checkout process.	
Basic Course of Action	**User Action**	**System Response**
	1. The customer accesses their shopping cart page or clicks on the "Cart" icon/button to view the contents of their shopping cart. 3. The customer identifies the item for which they want to update the quantity or options and modifies the selected item using an input field, plus/minus buttons, or a similar method provided by the system. 6. The customer may choose to continue browsing, add more items to the cart, or proceed to the checkout process.	2. The system displays the list of items in the customer's shopping cart, including their names, quantities, prices, and any relevant information. 4. The system validates the updated quantity or options, ensuring they are within the permissible limits and availability. 5. The system updates the shopping cart, displaying the updated number of items and the total value. 7. Use case Exit
Alternate Course of Action	4.1 If the customer tries to update the quantity to a value exceeding the available stock, the system can either limit the quantity to the available stock or notify the customer of the stock limitation. 4.1 If the customer attempts to update the quantity of an item to zero or remove all quantities, the system can prompt for confirmation to remove the item from the cart altogether.	

Use-case Register

Table 2.3: Use Case- Register

Use-case Number	UC-06	
Use-Case Name	Register	
Priority	High	
Actor/s	Customer	
Description	The customer wants to create a new account on the Future-Tech website.	
Precondition	The customer is not currently registered or logged in.	
Post-condition	• The customer has successfully registered a new account on the e-commerce website. • The customer can access their account and utilize additional features and functionalities	
Basic Course of Action	**User Action**	**System Response**
	1. The customer accesses the registration page. 2. The customer enters their personal information and submits the registration form. 5. The customer verifies their email address by clicking on the provided verification link. 7. The customer receives a confirmation message indicating successful registration. 8. The customer can now log into their account using their registered email address and password.	3. The system validates the information and creates a new customer account if it passes validation. 4. The system sends a verification email to the customer's provided email address. 6. The system activates the customer's account. 9. Use case Exit
Alternate Course of Action	4.1 If the provided email address is already associated with an existing account, the system prompts the customer to provide a different email address or prompts them to log in with their existing account. 4.2 If the provided password does not meet the required criteria, the system notifies the customer and provides guidelines for creating a valid password. 5.1 If the customer does not receive the verification email, the system provides an option to resend the verification email or contact customer support for assistance.	

Use-case Update Password

Table 2.3: Use Case- Update Password

Use-case Number	UC-07	
Use-Case Name	Update Password	
Priority	High	
Actor/s	Customer	
Description	The customer wants to update the password for their account on the Future-Tech website.	
Precondition	UC-01	
Post-condition	• The customer has successfully updated the password for their account on the e-commerce website. • The customer can log into their account using the new password for future access.	
Basic Course of Action	**User Action**	**System Response**
	1. The customer accesses their account settings or profile page. 2. The customer selects the option to update their password. 4. The customer enters their current password. 7. The customer enters their new desired password and confirms it by entering it again.	3. The system prompts the customer to enter their current password as a security measure. 5. The system verifies the entered password against the customer's current password stored in the system. 6. If the entered password matches the current password, the system allows the customer to proceed with updating their password. 8. If the new password passes validation, the system updates the customer's account with the new password. 9. The system displays a confirmation message to the customer, indicating that their password has been successfully updated. 10. Use case Exit
Alternate Course of Action	5. If the entered current password does not match the customer's current password, the system notifies the customer and prompts them to enter the correct password. 5.2 If the customer forgets their current password, the system can provide an option to reset the password using a password reset mechanism (e.g., sending a reset link to the customer's registered email address). 8. If the entered new password does not meet the required criteria, the system notifies the customer and provides guidelines for creating a valid password.	

Use-case User Profile Update

Table 2.3: Use Case- User Profile Update

Use-case Number	UC-08	
Use-Case Name	User Profile Update	
Priority	High	
Actor/s	Customer	
Description	The customer wants to update their profile information on the Future-Tech website.	
Precondition	UC-01 The customer has accessed their profile or account settings page.	
Post-condition	• The customer has successfully updated their profile information on the e-commerce website. • The customer's updated profile information is reflected and displayed correctly on their profile or account settings page.	
Basic Course of Action	**User Action**	**System Response**
	1. The customer accesses their account settings or profile page. 2. The customer updates the desired fields in their profile. 3. The customer submits the updated profile information. 6. The customer can now view their updated profile information in their account settings.	4. The system validates the information and updates the customer's profile if it passes validation. 5. The system displays a confirmation message indicating a successful profile update. 7. Use case Exit
Alternate Course of Action	4.1 If the modified information fails validation, the system notifies the customer of the specific validation errors or required corrections. 4.2 If the customer wants to update their password, they may need to follow the separate "Password Update" use case.	

Use-case User Checkout

Table 2.3: Use Case- Check Out

Use-case Number	UC-9	
Use-Case Name	Checkout	
Priority	High	
Actor/s	Customer	
Description	The customer wants to complete the purchase process and proceed to the finalization of the order on the Future-Tech website.	
Precondition	UC-01 UC-04	
Post-condition	• The customer has successfully completed the check-out process on the e-commerce website. • The order status is updated as "Placed" or "Processing," and the system generates an order confirmation. • The customer receives the order confirmation and can proceed with any further necessary actions.	
Basic Course of Action	**User Action**	**System Response**
	1. The customer proceeds to the checkout page or selects the "Checkout" option from the shopping cart. 3. The customer reviews the item details, ensuring accuracy and making any necessary adjustments. 5. The customer enters the necessary information accurately into the corresponding fields.	2. The system presents a summary of the selected items, including the names, quantities, and prices. 4. The system prompts the customer to provide the required information for the purchase, such as shipping address, contact details, and preferred shipping method. 6. The system validates the provided information for completeness and correctness. 7. If any errors or missing information are detected, the system prompts the customer to correct or complete the required fields. 8. Use case Exit
Alternate Course of Action	3.1 If the customer decides to modify the selected items before proceeding with the checkout: a. The customer navigates back to the shopping cart to remove or add items as needed. b. The system updates the shopping cart and recalculates the total purchase amount accordingly. c. The customer returns to the checkout process once the changes are made. 3.2 If the customer chooses to cancel the checkout process: a. The customer selects the cancel option or navigates away from the checkout page. b. The system updates the shopping cart and cancels the checkout process, returning the customer to the previous page.	

Use-case Add shipping info

Table 2.3: Use Case- Add shipping info

Use-case Number	UC-10	
Use-Case Name	Add shipping info	
Priority	High	
Actor/s	Customer	
Description	The customer wants to provide shipping information for their order on the Future-Tech website.	
Precondition	UC-01 UC-04 UC-09	
Post-condition	• The customer has successfully provided shipping information for their order on the Future-Tech website. • The shipping information is associated with the customer's order and will be used for order fulfillment and delivery purposes. • The customer can proceed with the remaining steps of the checkout process, such as selecting a payment method and confirming the order.	
Basic Course of Action	**User Action**	**System Response**
	1. The customer proceeds to the checkout page. 2. The customer enters the required shipping details, such as the shipping address. 5. The customer selects the desired shipping option. 6. The customer proceeds to the next step of the checkout process.	3. The system validates the entered shipping information. 4. If the shipping information passes validation, the system stores it in association with the customer's order. 7. Use case Exit
Alternate Course of Action	3. If the entered shipping address fails validation, the system notifies the customer of the specific error(s) and prompts them to correct the information accordingly.	

Use-case User Payment

Table 2.3: Use Case- User Payment

Use-case Number	UC-11	
Use-Case Name	Payment	
Priority	High	
Actor/s	Customer	
Description	The customer wants to make a payment for their selected items on the Future-Tech website.	
Precondition	UC-01 UC-04 UC-09	
Post-condition	• The customer has successfully made a payment for their order on the Future-Tech website. • The order status is updated as "Paid," and the system generates an order confirmation. • The customer receives the order confirmation and can proceed with any further necessary actions.	
Basic Course of Action	**User Action**	**System Response**
	1. The customer proceeds to the checkout page. 3. The customer selects a payment method and enters the required payment details. 6. The customer receives the order confirmation and can proceed with further actions,	2. The system displays a summary of the customer's order. 4. The system securely processes the payment using the provided information. 5. If the payment is successful, the system updates the order status as "Paid" and generates an order confirmation. 7. Use case Exit
Alternate Course of Action	4. If the payment authorization fails, the system notifies the customer and provides instructions or alternative payment methods to complete the transaction. 6. If the customer decides to cancel the payment, they can choose to return to the shopping cart or continue browsing without completing the transaction.	

Logout

Table 2.0: Use Case – Logout

Use-Case Number	UC-12	
Use-Case Name	Log out	
Priority	High	
Actor	Customer, Admin, Manager, stock keeper, sales officer	
Description	These use case allow Customers and Staff to log out from the system at the time of accomplishing their work.	
Precondition	UC-1	
Post Condition	System logs out	
Basic Course of Action	**User Action**	**System Response**
	1. The Customers or Staff want to log out 2. The Customer or Staff clicks the logout button.	3. The system responds to the requested action. 4. The system displays a message that the Customer or Staff is out of the system. 5. Use case Ends

Use-case User Create Account

Table 2.3: Use Case- Create Account

Use-case Number	UC-13	
Use-Case Name	Create Account	
Priority	High	
Actor/s	Admin	
Description	The admin wants to create a new account for an administrator on the system.	
Precondition	UC-01 The admin has the necessary permissions and access rights to create new accounts.	
Post-condition	• The admin has successfully created a new user account in the system. • The new account is stored in the system's database or user repository and can be used for authentication and access control purposes. • The admin can proceed with any further necessary actions, such as managing or modifying the created account.	
Basic Course of Action	**User Action**	**System Response**
	1. The admin accesses the user management section. 2. The admin selects the option to create a new user account. 4. The admin enters the necessary information for the new user account. 11. The admin can proceed with any further necessary actions, such as notifying the user of their account creation or granting additional privileges. 12. Use case Exit	3. The system presents a form for the admin to enter the required user details, such as username, password, email address, and user role. 5. The system validates the entered information, ensuring it meets any required criteria. 6. If the entered information passes validation, the system creates a new user account with the provided details. 7. The system generates a unique identifier (e.g., user ID) for the new account. 8. The system assigns the appropriate user role to the new account based on the admin's selection or default settings. 9. The system saves the new user account in the system's database or user repository. 10. The system displays a confirmation message to the admin, indicating that the new account has been successfully created.
Alternate Course of Action	5. If the entered information does not pass validation, the system notifies the admin of the specific error(s) and prompts them to correct the information accordingly.	

Use-case User Update Account

Table 2.3: Use Case- Update Account

<table>
<tr><td>Use-case Number</td><td colspan="2">UC-14</td></tr>
<tr><td>Use-Case Name</td><td colspan="2">Update Account</td></tr>
<tr><td>Priority</td><td colspan="2">High</td></tr>
<tr><td>Actor/s</td><td colspan="2">Admin</td></tr>
<tr><td>Description</td><td colspan="2">The admin wants to update the details of a user account in the system.</td></tr>
<tr><td>Precondition</td><td colspan="2">UC-01
UC-13</td></tr>
<tr><td>Post-condition</td><td colspan="2">• The admin has successfully updated the details of the user account in the system.
• The updated account information is stored in the system's database or user repository.
• The admin can proceed with any further necessary actions, such as managing or reviewing the updated account.</td></tr>
<tr><td rowspan="2">Basic Course of Action</td><td><b>User Action</b></td><td><b>System Response</b></td></tr>
<tr><td>1. The admin accesses the user management section or account management module.
2. The admin selects the user account they want to update.
4. The admin modifies the desired fields or attributes of the user account, such as username, email address, or user role.
11. The admin can proceed with any further necessary actions, such as notifying the user of the account update or reviewing the updated account details.
12. Use case Exit</td><td>3. The system retrieves the current details of the selected user account.
5. The system validates the entered information, ensuring it meets any required criteria
6. If the entered information passes validation, the system updates the user account with the provided details.
7. The system saves the updated user account in the system's database or user repository.
8. The system displays a confirmation message to the admin, indicating that the account has been successfully updated.</td></tr>
<tr><td>Alternate Course of Action</td><td colspan="2">5. If the entered information does not pass validation, the system notifies the admin of the specific error(s) and prompts them to correct the information accordingly.</td></tr>
</table>

Use-case Reset Password (Admin)

Table 2.3: Use Case- Reset Password (Admin)

Use-case Number	UC-15	
Use-Case Name	Reset Password (Admin)	
Priority	Low	
Actor/s	Admin	
Description	The admin wants to update the details of a user account in the system.	
Precondition	UC-01 UC-13	
Post-condition	• The admin has successfully reset the password for the user account in the system. • The user account can now be accessed using the new password.	
Basic Course of Action	**User Action**	**System Response**
	1. The admin accesses the user management section or account management module. 2. The admin selects the user account for which they want to reset the password. 4. The admin initiates the password reset process for the user account. 8. The admin confirms the completion of the password reset process. 9. Use case Exit	3. The system retrieves the details of the selected user account. 5. The system generates a temporary or random password for the user account. 6. The system securely stores the new password, replacing the existing password for the user account. 7. The system sends a notification or email to the user, informing them about the password reset and providing the new temporary password.
Alternate Course of Action	3.1 If the admin encounters any issues during the password reset process (e.g., user account not found, communication failure), the system notifies the admin and provides instructions for resolving the issue. 7.1 If the user fails to receive or use the temporary or one-time password within a specified timeframe, the admin may need to initiate the password reset process again.	

Use-case Delete User Account (Admin)

Table 2.3: Use Case- Delete User Account (Admin)

<table>
<tr><td>Use-case Number</td><td colspan="2">UC-16</td></tr>
<tr><td>Use-Case Name</td><td colspan="2">Delete User Account (Admin)</td></tr>
<tr><td>Priority</td><td colspan="2">Low</td></tr>
<tr><td>Actor/s</td><td colspan="2">Admin</td></tr>
<tr><td>Description</td><td colspan="2">The admin wants to delete a user account from the system.</td></tr>
<tr><td>Precondition</td><td colspan="2">UC-01
UC-13</td></tr>
<tr><td>Post-condition</td><td colspan="2">• The admin has successfully deleted the user account from the system.
• All associated data and records related to the user account have been removed from the system's database or user repository.
• The admin can proceed with any further necessary actions, such as updating any related user records or notifying other stakeholders about the account deletion.</td></tr>
<tr><td rowspan="2">Basic Course of Action</td><td><b>User Action</b></td><td><b>System Response</b></td></tr>
<tr><td>1. The admin accesses the user management section or account management module.
2. The admin selects the user account they want to delete.
4. The admin confirms their intention to delete the user account.
8. The admin can proceed with any further necessary actions, such as updating any related user records or notifying other stakeholders about the account deletion.
9. Use case Exit</td><td>3. The system displays a confirmation dialog to ensure the admin's intention to delete the user account.
5. The system removes all associated data and records related to the user account from the system.
6. The system updates the system's database or user repository to reflect the deletion of the user account.
7. The system sends a notification to the admin, confirming the successful deletion of the user account.</td></tr>
<tr><td>Alternate Course of Action</td><td colspan="2">5.1 If there are dependencies or restrictions preventing the deletion of the user account (e.g., ongoing orders, pending transactions), the system notifies the admin of these dependencies and prompts them to resolve or mitigate them before proceeding with the deletion.
4.1 If the admin decides to cancel the deletion request after confirming it, they can abort the deletion process, and the user account remains intact.</td></tr>
</table>

Use-case Supplier Registration (Manager)

Table 2.3: Use Case- Supplier Registration (Manager)

<table>
<tr><td>Use-case Number</td><td colspan="2">UC-17</td></tr>
<tr><td>Use-Case Name</td><td colspan="2">Supplier Registration (Manager)</td></tr>
<tr><td>Priority</td><td colspan="2">Low</td></tr>
<tr><td>Actor/s</td><td colspan="2">Manager</td></tr>
<tr><td>Description</td><td colspan="2">The manager wants to register a new supplier in the system.</td></tr>
<tr><td>Precondition</td><td colspan="2">The manager is logged into the system as an authorized user with managerial privileges.
The supplier to be registered does not already exist in the system.</td></tr>
<tr><td>Post-condition</td><td colspan="2">• The manager has successfully registered a new supplier in the system.
• The new supplier profile is stored in the system's database or supplier repository and can be used for supplier management and communication purposes.
• The manager can proceed with any further necessary actions, such as managing or reviewing the registered supplier.</td></tr>
<tr><td rowspan="2">Basic Course of Action</td><td><b>User Action</b></td><td><b>System Response</b></td></tr>
<tr><td>1. The manager accesses the supplier management section or supplier registration module.
2. The manager selects the option to register a new supplier.
4. The manager enters the required information for the new supplier.
8. The admin can proceed with any further necessary actions, such as updating any related user records or notifying other stakeholders about the account deletion.
10. The manager can proceed with any further necessary actions, such as notifying the supplier of their registration or managing the supplier's details in the system.
11. Use case Exit</td><td>3. The system presents a form for the manager to enter the supplier's information, such as name, contact details, address, and relevant business information.
5. The system validates the entered information, ensuring it meets any required criteria
6. If the entered information passes validation, the system creates a new supplier record with the provided details.
7. The system assigns a unique identifier (e.g., supplier ID) to the new supplier.
8. The system saves the new supplier record in the system's supplier database or repository.
9. The system displays a confirmation message to the manager, indicating that the new supplier has been successfully registered.</td></tr>
<tr><td>Alternate Course of Action</td><td colspan="2">5.1 If the entered information does not pass validation, the system notifies the manager of the specific error(s) and prompts them to correct the information accordingly.
5.2 If there are additional fields or attributes required for supplier registration (e.g., tax identification number, product categories), the manager provides the necessary information accordingly.</td></tr>
</table>

Use-case Add Products (Manager)

Table 2.3: Use Case- Add Products (Manager)

Use-case Number	UC-18	
Use-Case Name	Add Products (Manager)	
Priority	Low	
Actor/s	Manager	
Description	The manager wants to add new products to the system's inventory.	
Precondition	The manager is logged into the system as an authorized user with managerial privileges. The system's inventory management module is accessible.	
Post-condition	• The manager has successfully added a new product to the system's inventory. • The new product listing is stored in the system's database or inventory repository and can be used for product management, ordering, and sales purposes. • The manager can proceed with any further necessary actions, such as managing or reviewing the added product.	
Basic Course of Action	**User Action**	**System Response**
	1. The manager accesses the product management section or product inventory module. 2. The manager selects the option to add a new product. 4. The manager enters the necessary information for the new product. 10. The manager can proceed with any further necessary actions, such as reviewing the product details, setting up pricing or discounts, or associating the product with relevant categories or suppliers. 11. Use case Exit	3. The system presents a form for the manager to enter the details of the new product, such as product name, description, price, quantity, and any other relevant attributes. 5. The system validates the entered information, ensuring it meets any required criteria. 6. If the entered information passes validation, the system creates a new product listing with the provided details. 7. The system generates a unique identifier (e.g., product ID) for the new product. 8. The system saves the new product listing in the system's database or product repository. 9. The system displays a confirmation message to the manager, indicating that the new product has been successfully added to the inventory.
Alternate Course of Action	5.1 If the entered information does not pass validation, the system notifies the manager of the specific error(s) and prompts them to correct the information accordingly. 5.2 If there are additional fields or attributes required for product listing (e.g., product images, product variants), the manager provides the necessary information accordingly.	

Use-case Update Products (Manager)

Table 2.3: Use Case- Update Products (Manager)

<table>
<tr><td>Use-case Number</td><td colspan="2">UC-19</td></tr>
<tr><td>Use-Case Name</td><td colspan="2">Update Products (Manager)</td></tr>
<tr><td>Priority</td><td colspan="2">Low</td></tr>
<tr><td>Actor/s</td><td colspan="2">Manager</td></tr>
<tr><td>Description</td><td colspan="2">The manager wants to update the details of existing products in the system's inventory.</td></tr>
<tr><td>Precondition</td><td colspan="2">The manager is logged into the system as an authorized user with managerial privileges.
The system's inventory management module is accessible.
The product(s) to be updated exist in the system's inventory.</td></tr>
<tr><td>Post-condition</td><td colspan="2">• The manager has successfully updated the details of the product in the system's inventory.
• The updated product listing is stored in the system's database or inventory repository and reflects the changes made by the manager.
• The manager can proceed with any further necessary actions, such as managing or reviewing the updated product.</td></tr>
<tr><td>Basic Course of Action</td><td><b>User Action</b></td><td><b>System Response</b></td></tr>
<tr><td></td><td>1. The manager accesses the inventory management section or product management module.
2. The manager selects the option to update an existing product.
4. The manager selects the product they want to update.
6. The manager modifies the desired fields or attributes of the product, such as product name, description, price, quantity, and any relevant attributes.
11. The manager can proceed with any further necessary actions, such as reviewing the updated product details, adjusting pricing or discounts, or associating the product with relevant categories or suppliers.
12. Use case Exit</td><td>3. The system presents a list of available products for the manager to choose from.
5. The system retrieves the current details of the selected product.
7. The system validates the entered information, ensuring it meets any required criteria.
8. If the entered information passes validation, the system updates the product listing with the provided details.
9. The system saves the updated product listing in the system's database or inventory repository.
10. The system displays a confirmation message to the manager, indicating that the product has been successfully updated.</td></tr>
<tr><td>Alternate Course of Action</td><td colspan="2">7.1 If the entered information does not pass validation, the system notifies the manager of the specific error(s) and prompts them to correct the information accordingly.
7.2 If there are additional fields or attributes required for the product update (e.g., product images, product variants), the manager provides the necessary information accordingly.</td></tr>
</table>

Use-case Search Products (Manager)

Table 2.3: Use Case- Search Products (Manager)

Use-case Number	UC-20	
Use-Case Name	Search Products (Manager)	
Priority	Low	
Actor/s	Manager	
Description	The manager wants to search for specific products within the system's inventory.	
Precondition	The manager is logged into the system as an authorized user with managerial privileges. The system's inventory management module is accessible	
Post-condition	• The manager has successfully searched for specific products within the system's inventory. • The search results, as well as the detailed information about the selected product(s), are displayed to the manager. • The manager can proceed with any further necessary actions, such as updating the selected product(s), generating reports, or performing additional searches.	
Basic Course of Action	**User Action**	**System Response**
	1. The manager accesses the inventory management section or product management module. 2. The manager selects the option to search for products. 4. The manager enters the search criteria, such as product name, category, supplier, or any relevant attributes. 7. The manager reviews the search results and selects the desired product(s) from the list. 9. The manager can proceed with any further necessary actions, such as updating the selected product(s), viewing sales history, or generating reports based on the search results. 10. Use case Exit	3. The system presents a search interface or search form for the manager to enter search criteria. 5. The system performs a search based on the entered criteria and retrieves the matching products. Successfully updated. 6. The system displays the search results, listing the products that match the search criteria. 8. The system displays detailed information about the selected product(s), including its attributes, price, quantity, and any associated data.
Alternate Course of Action	5. If no products match the entered search criteria, the system notifies the manager and allows them to refine the search criteria or try a different search approach.	

Use-case Remove Products (Manager)

Table 2.3: Use Case- Remove Products (Manager)

Use-case Number	UC-21	
Use-Case Name	Remove Products (Manager)	
Priority	Low	
Actor/s	Manager	
Description	The manager wants to remove a product from the system's inventory.	
Precondition	The manager is logged into the system as an authorized user with managerial privileges. The system's inventory management module is accessible. The product to be removed exists in the system's inventory	
Post-condition	• The manager has successfully removed the product from the system's inventory. • Any associated references or associations with the removed product are updated in the system. • The manager can proceed with any further necessary actions, such as managing or reviewing the updated product list.	
Basic Course of Action	**User Action**	**System Response**
	1. The manager accesses the inventory management section or product management module. 2. The manager selects the option to remove a product. 4. The manager selects the product they want to remove. 7. The manager confirms the removal request. 13. The manager can proceed with any further necessary actions, such as reviewing the updated product list or managing other products. 14. Use case Exit	3. The system presents a list of available products for the manager to choose from. 5. The system retrieves the details of the selected product. 6. The system presents a confirmation prompt to verify the manager's intent to remove the product. 8. The system validates any dependencies or restrictions associated with the product (e.g., existing orders, inventory levels). 9. If there are no dependencies or restrictions preventing the removal, the system proceeds with removing the product. 10. The system removes the product listing from the system's inventory. 11. The system updates any relevant references or associations to reflect the removal of the product. 12. The system displays a confirmation message to the manager, indicating that the product has been successfully removed from the inventory.
Alternate Course of Action	9.1 If there are dependencies or restrictions preventing the removal of the product (e.g., existing orders, inventory levels), the system notifies the manager of these dependencies and prompts them to resolve or mitigate them before proceeding with the removal.	

	7.1 If the manager decides to cancel the removal request after confirming it, they can abort the removal process, and the product remains in the inventory.

Use-case Enable and Disable Products (Manager)

Table 2.3: Use Case- Enable and Disable Products (Manager)

<table>
<tr><td>Use-case Number</td><td colspan="2">UC-22</td></tr>
<tr><td>Use-Case Name</td><td colspan="2">Enable and Disable Products (Manager)</td></tr>
<tr><td>Priority</td><td colspan="2">Low</td></tr>
<tr><td>Actor/s</td><td colspan="2">Manager</td></tr>
<tr><td>Description</td><td colspan="2">The manager wants to enable or disable a product within the system's inventory.</td></tr>
<tr><td>Precondition</td><td colspan="2">The manager is logged into the system as an authorized user with managerial privileges.
The system's inventory management module is accessible.
The product to be enabled or disabled exists in the system's inventory.</td></tr>
<tr><td>Post-condition</td><td colspan="2">• The manager has successfully enabled or disabled the availability of the product in the system's inventory.
• The availability status of the product is updated in the system's database or inventory repository.
• The manager can proceed with any further necessary actions, such as managing or reviewing the updated product list.</td></tr>
<tr><td rowspan="2">Basic Course of Action</td><td><b>User Action</b></td><td><b>System Response</b></td></tr>
<tr><td>1. The manager accesses the inventory management section or product management module.
2. The manager selects the option to enable or disable a product.
4. The manager selects the product they want to enable or disable.
7. The manager selects the desired action to enable or disable the product.
11. The manager can proceed with any further necessary actions, such as reviewing the updated product list or managing other products.
14. Use case Exit</td><td>3. The system presents a list of available products for the manager to choose from.
5. The system retrieves the details of the selected product.
6. The system displays the current availability status of the product (enabled or disabled).
8. The system updates the availability status of the product accordingly.
9. The system saves the changes in the system's database or inventory repository.
10. The system displays a confirmation message to the manager, indicating that the availability status of the product has been successfully updated.</td></tr>
<tr><td>Alternate Course of Action</td><td colspan="2">8.1 If the selected product is already enabled and the manager chooses to enable it again, the system displays a message indicating that the product is already enabled.
8.2 If the selected product is already disabled and the manager chooses to disable it again, the system displays a message indicating that the product is already disabled.</td></tr>
</table>

Use-case Purchasing Stocks (Stock Keeper)

Table 2.3: Use Case- Purchasing Stocks (Stock Keeper)

Use-case Number	UC-23	
Use-Case Name	Purchasing Stocks (Stock Keeper)	
Priority	Low	
Actor/s	Stock Keeper	
Description	The stock keeper wants to purchase additional stocks of products for the inventory.	
Precondition	The stock keeper is logged into the system as an authorized user with stock management privileges. The system's stock management module is accessible. The stock levels of certain products are below the desired threshold.	
Post-condition	• The stock keeper has successfully purchased additional stocks for the identified products. • The purchased stocks are recorded in the system's database or procurement system. • The stock levels in the inventory are updated to reflect the newly purchased stocks. • The stock keeper can proceed with any further necessary actions, such as managing the inventory or reviewing the updated stock levels.	
Basic Course of Action	**User Action**	**System Response**
	1. The stock keeper accesses the stock management section or inventory management module. 2. The stock keeper selects the option to purchase new stocks. 4 The stock keeper selects the products they need to restock or enters the product information, such as product name, quantity, and desired supplier. 6 The stock keeper enters the quantity of each product they want to purchase. 8. The stock keeper reviews and confirms the purchase order or request. 9. Since the supplier is not connected to the system, the stock keeper manually communicates the purchase order to the supplier through an alternative channel (e.g., email, phone, fax). 10. The stock keeper receives the supplier's response, which may include order confirmation, estimated delivery date, or any necessary adjustments. 11. The stock keeper updates the purchase order status in the system, indicating that it has been	3. The system presents a list of products with low stock levels or prompts the stock keeper to enter the products they wish to purchase. 5. The system verifies the product availability and prompts the stock keeper to enter the desired quantity to purchase. 7. The system generates a purchase order or request with the selected products and quantities.

	sent to the supplier and awaiting fulfillment. 12. Upon receiving the stock from the supplier, the stock keeper manually updates the inventory in the system to reflect the received quantities. 14. Use case Exit	
Alternate Course of Action	5.1 If the desired products are not available from the chosen supplier, the system notifies the stock keeper and suggests alternative suppliers. 9.1 If the stock keeper encounters any issues while communicating with the supplier, such as unavailability or changes in product availability or pricing, they can revise the order accordingly and update the purchase order details in the system.	

Use-case Add New Stocks (Stock Keeper)

Table 2.3: Use Case- Add New Stocks (Stock Keeper)

Use-case Number	UC-24	
Use-Case Name	Add New Stocks (Stock Keeper)	
Priority	Low	
Actor/s	Stock Keeper	
Description	The stock keeper wants to add new stocks to the inventory.	
Precondition	The stock keeper is logged into the system as an authorized user with stock management privileges. The system's stock management module is accessible. The stock keeper has received new stocks to be added to the inventory.	
Post-condition	• The stock keeper has successfully added the new stocks to the inventory. • The stock levels in the inventory are updated to reflect the added quantities. • Any relevant processes triggered by the stock addition have been executed. • The stock keeper can proceed with any further necessary actions, such as reviewing the updated inventory or managing other stock-related tasks.	
Basic Course of Action	**User Action**	**System Response**
	1. The stock keeper accesses the stock management section or inventory management module. 2. The stock keeper selects the option to add new stocks. 4 The stock keeper enters the details of the new stocks, ensuring accuracy and completeness. 11. The stock keeper can proceed with any further necessary actions, such as reviewing the updated inventory or managing other stock-related tasks. 12. Use case Exit	3. The system prompts the stock keeper to enter the details of the new stocks, such as product information (name, SKU, description), quantity, and any additional attributes. 5. The system verifies the entered information for any errors or inconsistencies. 6. If errors or inconsistencies are found, the system notifies the stock keeper and prompts them to correct the information. 7. Once the information is validated, the system updates the inventory with the new stocks. 8. The system adjusts the stock levels to reflect the added quantity of each product. 9. The system may trigger any relevant processes, such as updating the stock records, generating purchase orders if stock levels were replenished, or notifying other stakeholders of the new stocks. 10. The system confirms the successful addition of the new stocks and provides a confirmation message to the stock keeper.

Alternate Course of Action	5.1 If the stock keeper attempts to add stocks with incomplete or invalid information, the system prompts them to provide the missing or corrected details before proceeding. 5.2 If there are any constraints or limitations on adding new stocks (e.g., maximum stock capacity, restricted products), the system notifies the stock keeper and prompts them to address the constraints before adding the stocks.

Use-case Issue Stock (Stock Keeper)

Table 2.3: Use Case- Issue Stock (Stock Keeper)

Use-case Number	UC-25	
Use-Case Name	Issue Stock (Stock Keeper)	
Priority	Low	
Actor/s	Stock Keeper	
Description	The stock keeper wants to issue stock from the inventory for various purposes, such as sales, internal transfers, or customer orders.	
Precondition	The stock keeper is logged into the system as an authorized user with stock management privileges. The system's stock management module is accessible. The stock keeper has received a request or instruction to issue stock.	
Post-condition	• The stock keeper has successfully issued the stock from the inventory. • The stock levels in the inventory are updated to reflect the issued quantities. • Any relevant processes triggered by the stock issuance have been executed. • The stock keeper can proceed with any further necessary actions, such as packaging the stock item, updating shipment details, or managing other stock-related tasks.	
Basic Course of Action	**User Action**	**System Response**
	1. The stock keeper accesses the stock management section or inventory management module. 2. The stock keeper selects the option to issue stock. 4. The stock keeper selects the stock item to be issued or enters the relevant details. 6. The stock keeper enters the quantity of the stock item to be issued. 12. The stock keeper can proceed with any further necessary actions, such as packaging the stock item, updating shipment details, or managing other stock-related tasks. 13. Use case Exit	3. The system presents a list of available stock items or prompts the stock keeper to enter the details of the stock item to be issued, such as product information (name, SKU, description). 5. The system verifies the availability of the selected stock item and prompts the stock keeper to enter the quantity to be issued. 7. The system checks the availability of the requested quantity of the stock item. 8. If the requested quantity is available, the system updates the stock levels to reflect the issued quantity. 9. The system generates any necessary documentation or records related to the stock issue, such as an inventory adjustment record, sales order, or transfer order. 10. The system may trigger any relevant processes, such as updating sales or accounting records, notifying other stakeholders, or generating shipping labels. 11. The system confirms the successful issuance of the stock and provides a confirmation message to the stock keeper.

Alternate Course of Action	5.1 If the requested quantity of the stock item is not available, the system notifies the stock keeper and prompts them to adjust the quantity or select an alternative stock item. 5.2 If there are any constraints or limitations on issuing stock (e.g., restricted items, insufficient stock levels), the system notifies the stock keeper and prompts them to address the constraints before issuing the stock.	

Use-case Update Stocks (Stock Keeper)

Table 2.3: Use Case- Update Stocks (Stock Keeper)

<table>
<tr><td>Use-case Number</td><td colspan="2">UC-26</td></tr>
<tr><td>Use-Case Name</td><td colspan="2">Update Stocks (Stock Keeper)</td></tr>
<tr><td>Priority</td><td colspan="2">Low</td></tr>
<tr><td>Actor/s</td><td colspan="2">Stock Keeper</td></tr>
<tr><td>Description</td><td colspan="2">The stock keeper wants to update the information or attributes of existing stocks in the inventory.</td></tr>
<tr><td>Precondition</td><td colspan="2">The stock keeper is logged into the system as an authorized user with stock management privileges.
The system's stock management module is accessible.
The stock keeper has identified the need to update specific stock items in the inventory.</td></tr>
<tr><td>Post-condition</td><td colspan="2">• The stock keeper has successfully updated the information or attributes of the stock item.
• The stock item in the inventory reflects the modified information or attributes.
• Any relevant processes triggered by the stock update have been executed.
• The stock keeper can proceed with any further necessary actions, such as reviewing the updated stock item or managing other stock-related tasks.</td></tr>
<tr><td rowspan="2">Basic Course of Action</td><td><b>User Action</b></td><td><b>System Response</b></td></tr>
<tr><td>1. The stock keeper accesses the stock management section or inventory management module.
2. The stock keeper selects the option to update stocks.
4. The stock keeper selects the stock item to be updated or enters the relevant details.
6. The stock keeper modifies the desired information or attributes of the stock item, such as quantity, price, description, or other relevant details.
12. The stock keeper can proceed with any further necessary actions, such as reviewing the updated stock item or managing other stock-related tasks.
13. Use case Exit</td><td>3. The system presents a list of existing stock items or prompts the stock keeper to enter the details of the stock item to be updated, such as product information (name, SKU, description).
5. The system retrieves the current information and attributes of the selected stock item.
7. The system validates the entered changes for any errors or inconsistencies.
8. If errors or inconsistencies are found, the system notifies the stock keeper and prompts them to correct the information.
9. Once the changes are validated, the system updates the stock item with the modified information or attributes.
10. The system may trigger any relevant processes, such as updating stock records, adjusting stock levels, or notifying other stakeholders of the changes.
11. The system confirms the successful update of the stock item and provides a confirmation message to the stock keeper.</td></tr>
</table>

Alternate Course of Action	7.1 If the stock keeper attempts to update stock items with incomplete or invalid information, the system prompts them to provide the missing or corrected details before proceeding. 7.2 If there are any constraints or limitations on updating stock items (e.g., restricted attributes, dependencies), the system notifies the stock keeper and prompts them to address the constraints before applying the changes.

Use-case View Stock (Stock Keeper)

Table 2.3: Use Case- View Stock (Stock Keeper)

<table>
<tr><td>Use-case Number</td><td colspan="2">UC-27</td></tr>
<tr><td>Use-Case Name</td><td colspan="2">View Stock (Stock Keeper)</td></tr>
<tr><td>Priority</td><td colspan="2">Low</td></tr>
<tr><td>Actor/s</td><td colspan="2">Stock Keeper</td></tr>
<tr><td>Description</td><td colspan="2">The stock keeper wants to view the stock information in the inventory, such as stock levels, location, or attributes.</td></tr>
<tr><td>Precondition</td><td colspan="2">The stock keeper is logged into the system as an authorized user with stock management privileges.
The system's stock management module is accessible.</td></tr>
<tr><td>Post-condition</td><td colspan="2">• The stock keeper has successfully viewed the stock information in the inventory.
• The stock levels, location, and attributes of the displayed stock items are accurate and up-to-date.
• The stock keeper can proceed with any further necessary actions, such as updating the stock, issuing stock, or managing other stock-related tasks.</td></tr>
<tr><td rowspan="2">Basic Course of Action</td><td><b>User Action</b></td><td><b>System Response</b></td></tr>
<tr><td>1. The stock keeper accesses the stock management section or inventory management module.
2. The stock keeper selects the option to view stock.
4. The stock keeper selects the stock item to view or performs a search to locate the desired stock item(s).
6. The stock keeper can scroll through the list of stock items or navigate to different pages to view more stock items.
7. If necessary, the stock keeper can apply filters or sorting options to refine the displayed stock items based on specific criteria.
9. The stock keeper can review the stock information, including the stock levels, location, and attributes, to ensure accuracy and monitor stock availability.
10. The stock keeper can proceed with any further necessary actions, such as updating the stock, issuing stock, or managing other stock-related tasks.
11. Use case Exit</td><td>3. The system presents a list of available stock items or provides search options to locate the desired stock item(s), such as by product information (name, SKU, description) or stock location.
5. The system displays the stock information for the selected stock item(s), including stock levels, location, attributes, and any additional relevant details.
8. The system provides options for the stock keeper to further interact with the displayed stock information, such as viewing stock history, generating stock reports, or initiating stock adjustments.</td></tr>
</table>

Alternate Course of Action	5. If the stock keeper performs a search for a specific stock item and no results are found, the system notifies the stock keeper and prompts them to refine their search criteria.

Use-case Issue Product to Couriers (Stock Keeper, Sales Officer)

Table 2.3: Use Case- Issue Product to Couriers (Stock Keeper, Sales Officer)

Use-case Number	UC-28	
Use-Case Name	Issue Products to Couriers (Stock Keeper, Sales Officer)	
Priority	Low	
Actor/s	Stock Keeper, Sales Officer	
Description	The stock keeper or sales officer wants to issue products from the inventory to couriers for delivery to customers.	
Precondition	The stock keeper or sales officer is logged into the system as an authorized user with the necessary privileges. The system's inventory management module and courier management system are accessible. The customer orders are processed and ready for delivery.	
Post-condition	• The stock keeper or sales officer has successfully issued the products from the inventory to the couriers for delivery. • The stock levels in the inventory are updated to reflect the issued products. • The couriers receive the necessary instructions and documentation for delivery. • The stock keeper or sales officer can proceed with any further necessary actions, such as updating the order status, providing tracking information to customers, or managing other order fulfillment tasks.	
Basic Course of Action	**User Action**	**System Response**
	1. The stock keeper or sales officer accesses the inventory management section or order fulfillment module. 2. The stock keeper or sales officer selects the option to issue products to couriers for delivery. 4. The stock keeper or sales officer selects the customer order(s) that are ready for delivery. 7. The stock keeper or sales officer selects the courier(s) from the available options or adds new courier details if necessary. 9. The stock keeper or sales officer verifies the accuracy of the packing list or delivery manifest. 13. The stock keeper or sales officer confirms the successful issuance of the products to the couriers. 14. The stock keeper or sales officer may proceed with any further necessary actions, such as updating the order status, providing tracking information to customers, or managing other order fulfillment tasks.	3. The system presents a list of available customer orders or prompts the stock keeper or sales officer to search for specific orders based on order ID, customer name, or other relevant criteria. 5. The system verifies the availability of the ordered products in the inventory. 6. If all products in the selected orders are available, the system prompts the stock keeper or sales officer to select the courier(s) for delivery. 8. The system generates a packing list or delivery manifest for each courier, listing the products to be issued for delivery. 10. The system updates the inventory to reflect the issued products and adjusts the stock levels accordingly. 11. The system generates any necessary documentation, such as delivery labels, invoices, or shipping notifications. 12. The system notifies the couriers of the issued products and provides them with the necessary instructions and documentation for delivery

	15. Use case Exit	
Alternate Course of Action	5.1 If any of the ordered products are not available in the inventory, the system notifies the stock keeper or sales officer and prompts them to address the stock availability issue before proceeding. 5.2 If there are any constraints or limitations on issuing products to couriers (e.g., restricted items, specific delivery timeframes), the system applies appropriate rules and notifications to the stock keeper or sales officer.	

Use-case Register Couriers (Stock Keeper/Sales Officer)

Table 2.3: Use Case- Register Couriers (Stock Keeper/Sales Officer)

Use-case Number	UC-29	
Use-Case Name	Register Couriers (Stock Keeper/Sales Officer)	
Priority	Low	
Actor/s	Stock Keeper, Sales Officer	
Description	The stock keeper or sales officer wants to register new couriers into the system for managing and assigning delivery tasks.	
Precondition	The stock keeper or sales officer is logged into the system as an authorized user with the necessary privileges. The system's courier management module is accessible.	
Post-condition	• The stock keeper or sales officer has successfully registered a new courier in the system. • The new courier is assigned a unique identifier or tracking code for identification and tracking purposes. • The registered courier details are stored in the system's courier management module. • The stock keeper or sales officer can proceed with any further necessary actions, such as assigning delivery tasks to the registered courier or managing other courier-related tasks.	
Basic Course of Action	**User Action**	**System Response**
	1. The stock keeper or sales officer accesses the courier management section or courier registration module. 2. The stock keeper or sales officer selects the option to register new couriers. 4. The stock keeper or sales officer enters the relevant information of the new courier, such as name, contact details, delivery area, and any other required fields. 9. The stock keeper or sales officer can review the newly registered courier details for verification. 11. The stock keeper or sales officer can proceed with any further necessary actions, such as assigning delivery tasks to the registered courier or managing other courier-related tasks. 12. Use case Exit	3. The system presents a courier registration form, prompting the stock keeper or sales officer to enter the necessary details of the new courier. 5. The system validates the entered information for completeness and accuracy. 6. If any errors or missing information are detected, the system notifies the stock keeper or sales officer and prompts them to correct the details. 7. Once the information is validated, the system creates a new courier profile in the system. 8. The system assigns a unique identifier or tracking code to the registered courier for identification and tracking purposes. 10. The system confirms the successful registration of the new courier and provides a confirmation message to the stock keeper or sales officer.
Alternate Course of Action	5.1 If the stock keeper or sales officer attempts to register a courier with duplicate or conflicting information, the system notifies them and prompts them to resolve the conflicts before proceeding. 5.2 If there are any specific requirements or prerequisites for courier registration (e.g., valid licenses, background checks), the system applies appropriate checks and validations before completing the registration process.	

Use-case Showroom purchase (Sales Officer)

Table 2.3: Use Case- showroom purchase (Sales Officer)

<table>
<tr><td>Use-case Number</td><td colspan="2">UC-30</td></tr>
<tr><td>Use-Case Name</td><td colspan="2">showroom purchase (Sales Officer)</td></tr>
<tr><td>Priority</td><td colspan="2">Low</td></tr>
<tr><td>Actor/s</td><td colspan="2">Stock Keeper</td></tr>
<tr><td>Description</td><td colspan="2">The sales officer wants to assist customers with their purchases in a showroom and facilitate the sales process.</td></tr>
<tr><td>Precondition</td><td colspan="2">The sales officer is present in the showroom and ready to assist customers.
The sales officer is logged into the system as an authorized user with the necessary privileges.
The system's point-of-sale (POS) or sales management module is accessible.</td></tr>
<tr><td>Post-condition</td><td colspan="2">• The sales officer has successfully assisted the customer with their purchase in the showroom.
• The sales transaction and payment details are recorded in the system.
• The customer leaves the showroom with the purchased products and the necessary documentation.
• The sales officer can proceed with any further necessary actions, such as updating the inventory, generating sales reports, or assisting other customers.</td></tr>
<tr><td rowspan="2">Basic Course of Action</td><td><b>User Action</b></td><td><b>System Response</b></td></tr>
<tr><td>1. The sales officer selects the desired products on behalf of the customer.
2. The sales officer enters the selected products and quantities into the system.
5. Use case Exit</td><td>3. The system checks the availability of the selected products.
4. If the products are available, the system calculates the total purchase amount and display.</td></tr>
<tr><td>Alternate Course of Action</td><td colspan="2">3.1 If any of the selected products are unavailable in the inventory, the sales officer informs the customer and suggests alternative products or arranges for the unavailable products to be ordered or restocked.
3.2 If the customer requests customization or special orders, the sales officer follows appropriate procedures to fulfill their requirements.</td></tr>
</table>

Use-case Register Customers (Sales Officer)

Table 2.3: Use Case- Register Customers (Sales Officer)

Use-case Number	UC-31	
Use-Case Name	Register Customers (Sales Officer)	
Priority	Low	
Actor/s	Sales Officer	
Description	The sales officer wants to register new customers into the system to facilitate future interactions and maintain customer records.	
Precondition	The sales officer is logged into the system as an authorized user with the necessary privileges. The system's customer management module is accessible.	
Post-condition	• The sales officer has successfully registered the customer into the system. • The customer's information is stored securely in the system's customer database. • The customer is provided with any relevant documentation, membership details, or other materials. • The sales officer can access and update the customer's information as needed for future interactions and transactions.	
Basic Course of Action	**User Action**	**System Response**
	1. The sales officer initiates the customer registration process in the system. 2. The sales officer requests the necessary information from the customer, such as their name, contact details, address, and any additional relevant data. 3. The sales officer enters the customer's information into the system, ensuring accuracy and completeness. 7. The sales officer provides the customer with their customer ID or any relevant membership or loyalty program details. 8. The sales officer provides the customer with a warm welcome, expresses gratitude for their registration, and assures them of personalized assistance and support. 9. Use case Exit	4. The system validates the entered information, checking for any duplicate or missing data. 5. If any errors or inconsistencies are found, the system alerts the sales officer, who then corrects the information accordingly. 6. The system generates a unique customer identifier or customer ID for the newly registered customer.
Alternate Course of Action	4.1 If the customer is already registered in the system, The sales officer verifies if the customer has an existing account. If the customer has an account, the sales officer informs them that they are already registered. The sales officer may update the customer's information if necessary or proceed with the current transaction. 4.2 If the customer provides incorrect or incomplete information, The sales officer verifies the provided information with the customer for accuracy and completeness. The sales officer requests the necessary corrections or missing details. Then sales officer re-enters the corrected information into the system.	

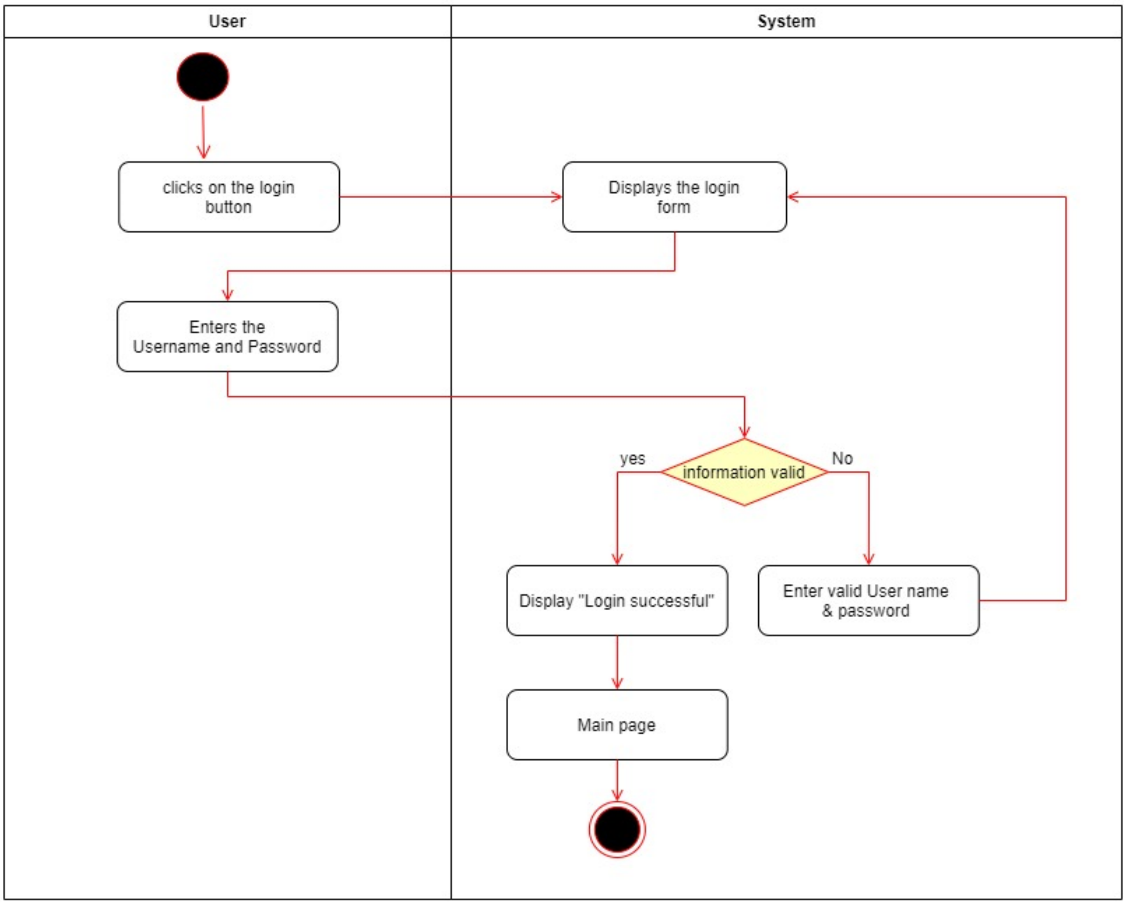
Use-case Receive Payment (Sales Officer)

Table 2.3: Use Case- Receive Payment (Sales Officer)

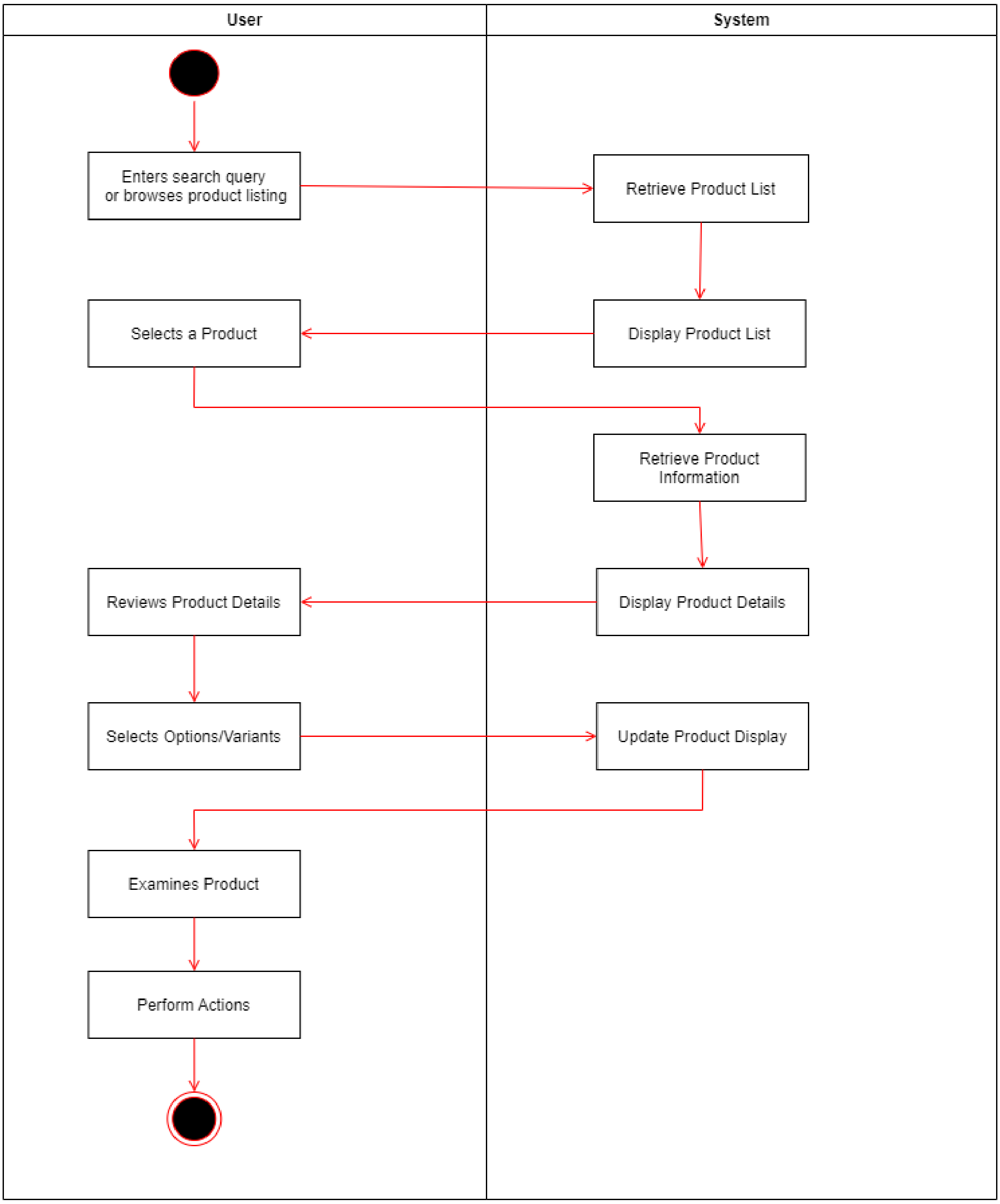
Use-case Number	UC-32	
Use-Case Name	Receive Payment (Sales Officer)	
Priority	Low	
Actor/s	Stock Keeper	
Description	The sales officer wants to accurately process and record payments from customers for their purchases.	
Precondition	The sales officer is logged into the system as an authorized user with the necessary privileges. The customer has completed their purchase and is ready to make the payment.	
Post-condition	• The sales officer has successfully received and processed the payment from the customer. • The payment details are recorded in the system, including the payment method, amount, and transaction reference number. • The sales transaction is marked as complete and closed in the system. • The customer receives a payment confirmation or receipt for their records. • The sales officer can proceed with any further necessary actions, such as updating inventory, generating sales reports, or assisting other customers.	
Basic Course of Action	**User Action**	**System Response**
	1. The sales officer requests payment from the customer for the total purchase amount. 3. The sales officer enters the payment method (cash, credit/debit card, mobile payment, etc.) into the system. 4. If the payment method is cash, the sales officer enters the cash amount received from the customer into the system. 5. If the payment method is non-cash (e.g., card, mobile payment), the sales officer initiates the payment process through the system or a payment terminal. 11. The sales officer confirms the successful payment to the customer and provides the payment confirmation or receipt. 12. The sales officer thanks the customer for their payment and confirms the completion of the transaction. 13. The sales officer may perform any additional tasks associated with the completed payment, such as updating sales reports or	2. The system prompts the sales officer to enter the payment details, such as the payment method and amount. 6. The system verifies and processes the payment, ensuring it is authorized and meets any validation criteria. 7. Once the payment is successfully processed, the system updates the sales transaction with the payment details, including the payment method, amount, and transaction reference number. 8. The system generates a payment confirmation or receipt, which the sales officer can provide to the customer. 9. The system updates relevant inventory or stock information based on the completed payment. 10. The system notifies the sales officer of the successful payment processing and provides any necessary additional information or prompts. 14. The system marks the sales transaction as complete and closed, allowing for any necessary post-transaction activities or reporting.

	initiating further actions based on the payment status. 15. Use case Exit	
Alternate Course of Action	6.1 If the customer provides an insufficient payment amount: a. The sales officer informs the customer of the shortfall and requests the remaining payment. b. The sales officer provides options for making up the difference, such as using an alternative payment method or removing items from the purchase. c. The sales officer proceeds with the payment once the correct amount is received. 6.2 If the customer's payment method is declined or encounters an error: a. The sales officer informs the customer about the issue with the payment transaction. b. The sales officer offers assistance in resolving the problem, such as suggesting an alternative payment method or contacting the payment service provider. c. The sales officer ensures that the sales transaction is not finalized until the payment is successfully processed. 6.3 If the customer requests a split payment: a. The sales officer accommodates the request by accepting multiple payment methods for different portions of the total purchase amount. b. The sales officer records the details of each payment method used and ensures the total amount received matches the total purchase amount.	

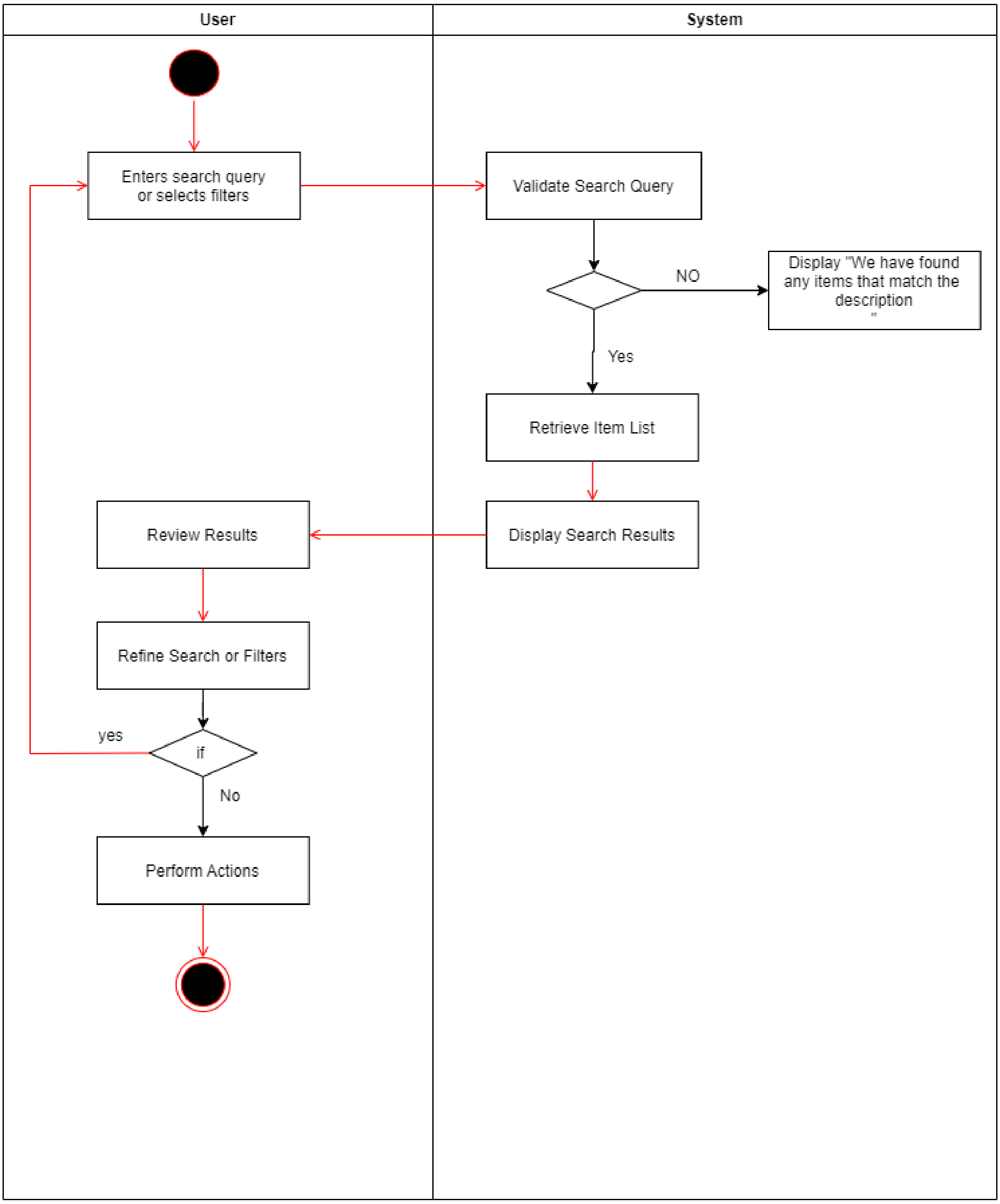
Login Activity diagrams


User
System
clicks on the login button
Displays the login form
Enters the Username and Password
information valid
yes
No
Display "Login successful"
Enter valid User name & password
Main page

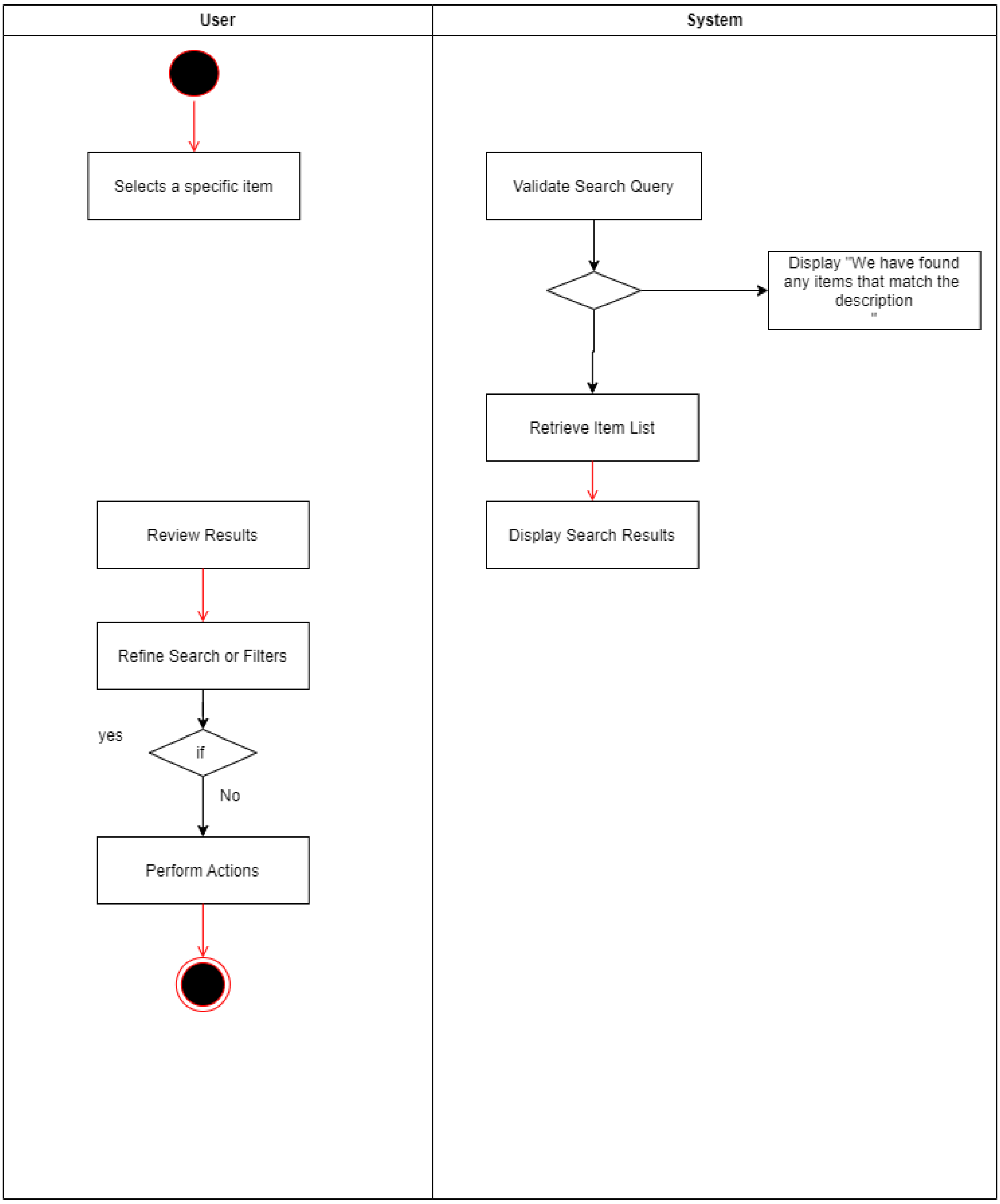
View items Activity diagrams



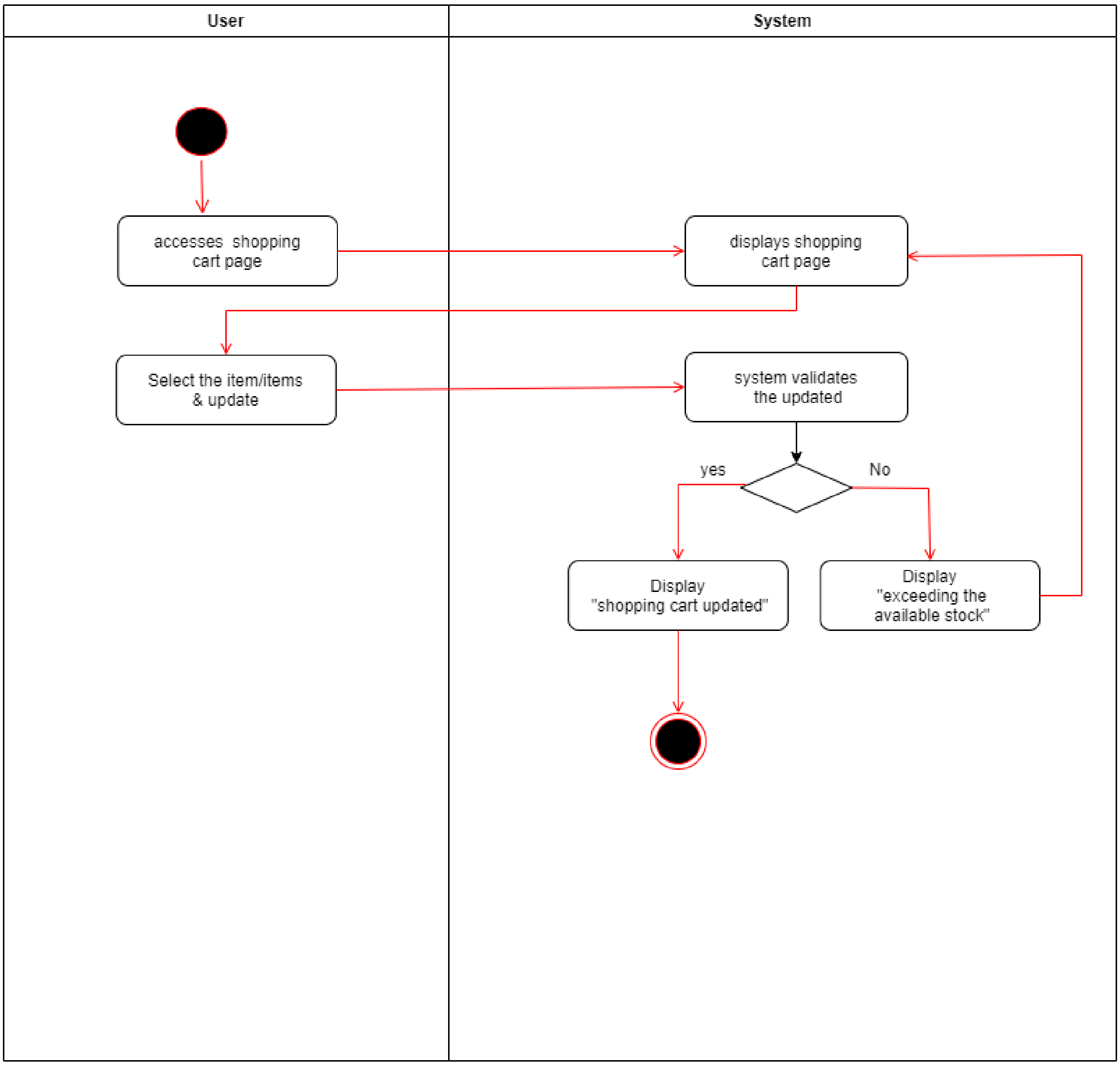
Search items Activity diagrams



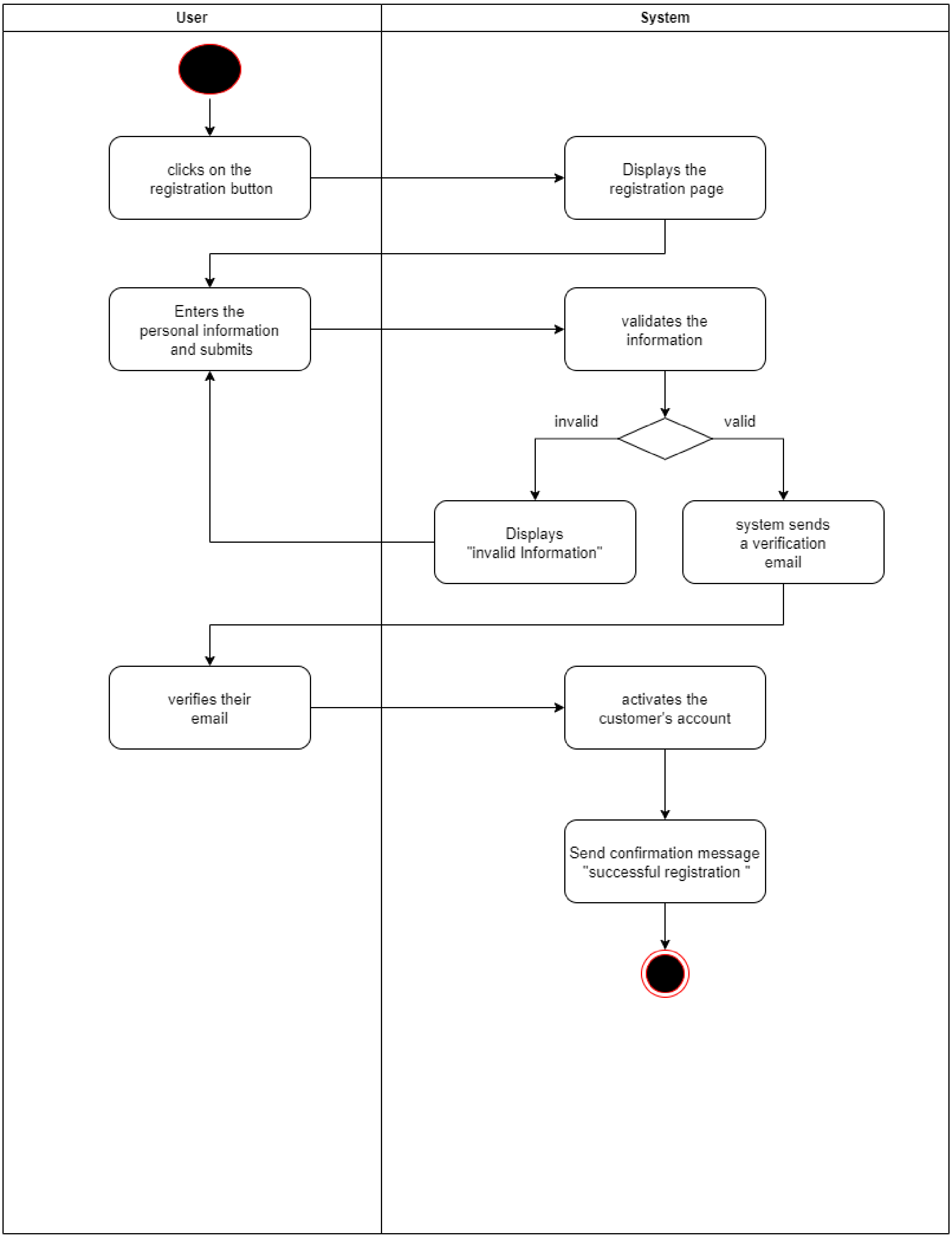
Add to cart Activity diagrams



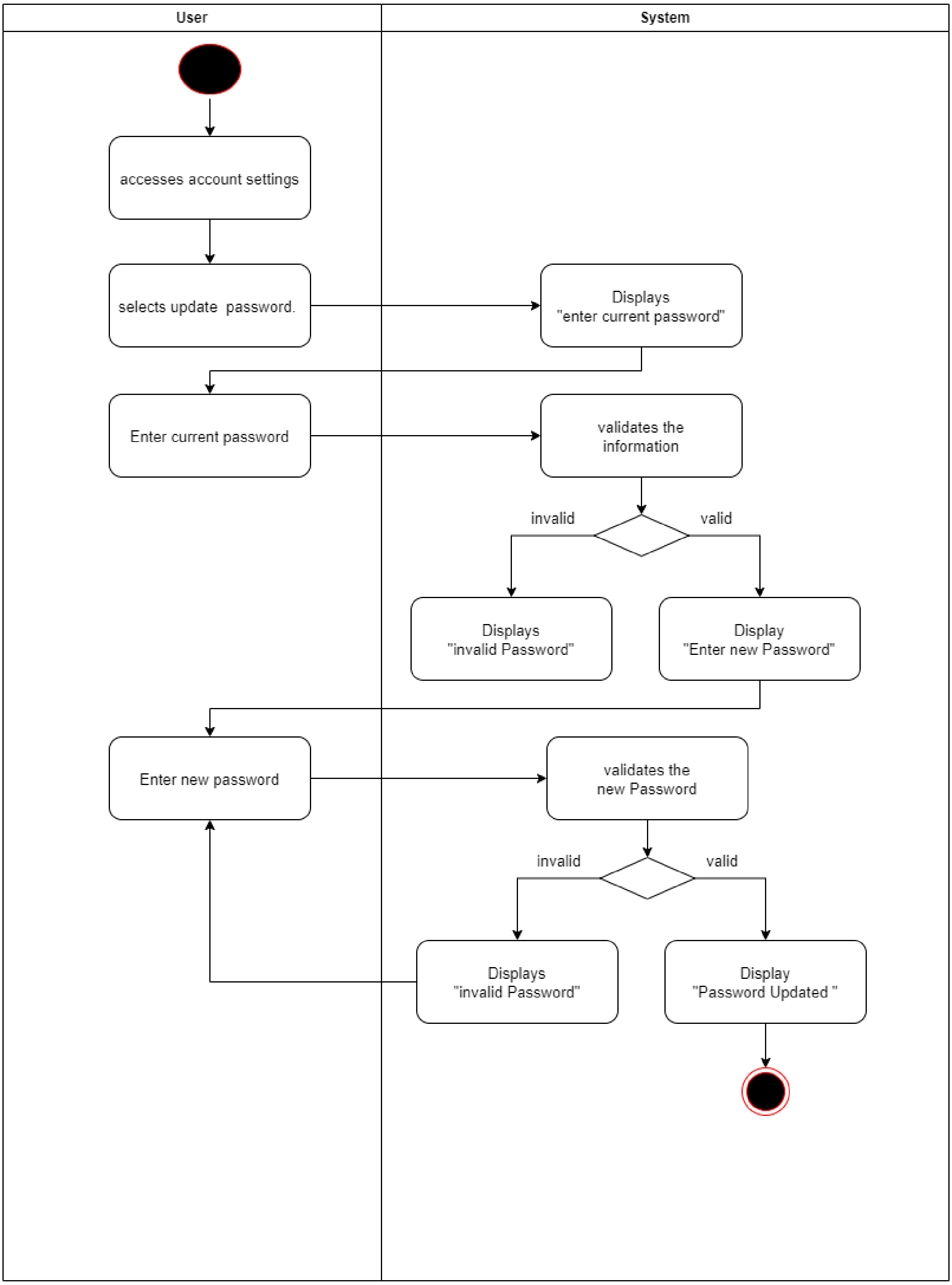
Update cart Activity diagrams


User
System
accesses shopping cart page
displays shopping cart page
Select the item/items & update
system validates the updated
yes
No
Display "shopping cart updated"
Display "exceeding the available stock"

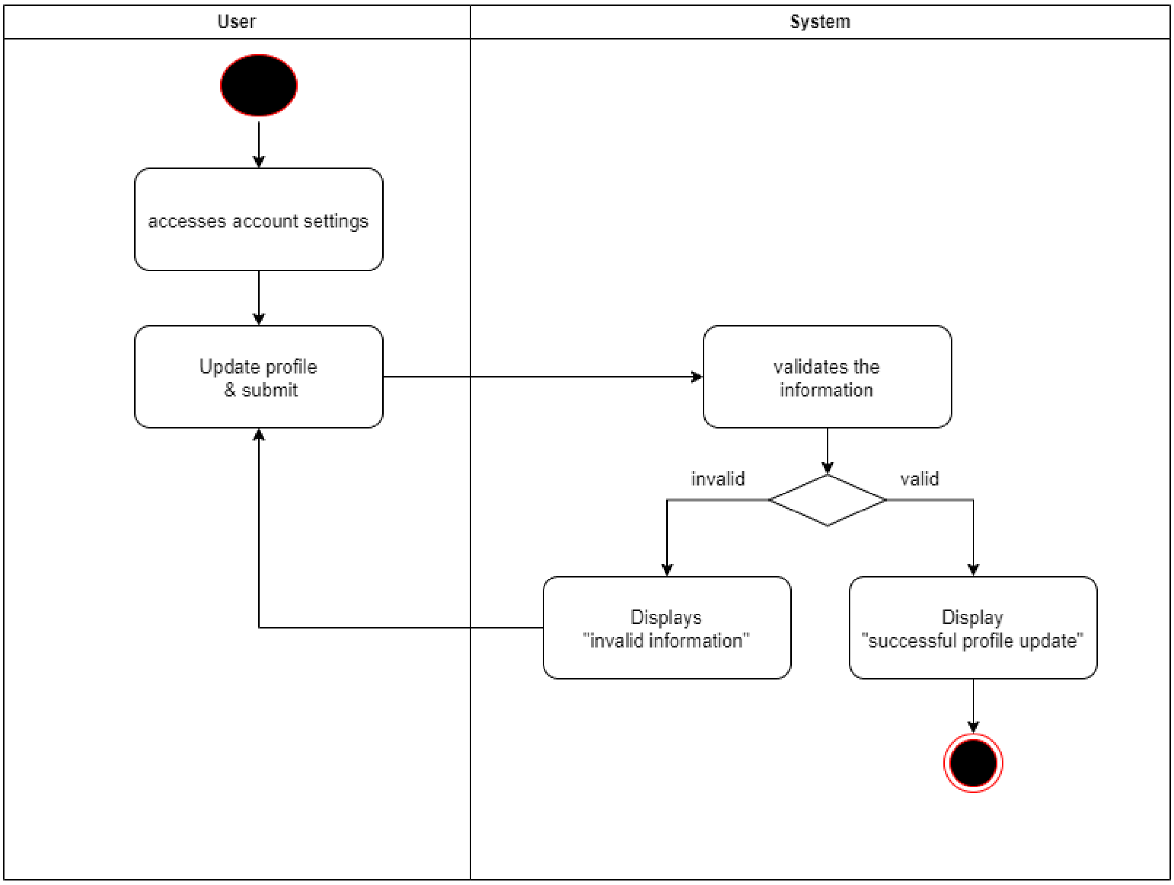
Register Customer Activity diagrams


User
System
clicks on the registration button
Displays the registration page
Enters the personal information and submits
validates the information
invalid
valid
Displays "invalid Information"
system sends a verification email
verifies their email
activates the customer's account
Send confirmation message "successful registration "

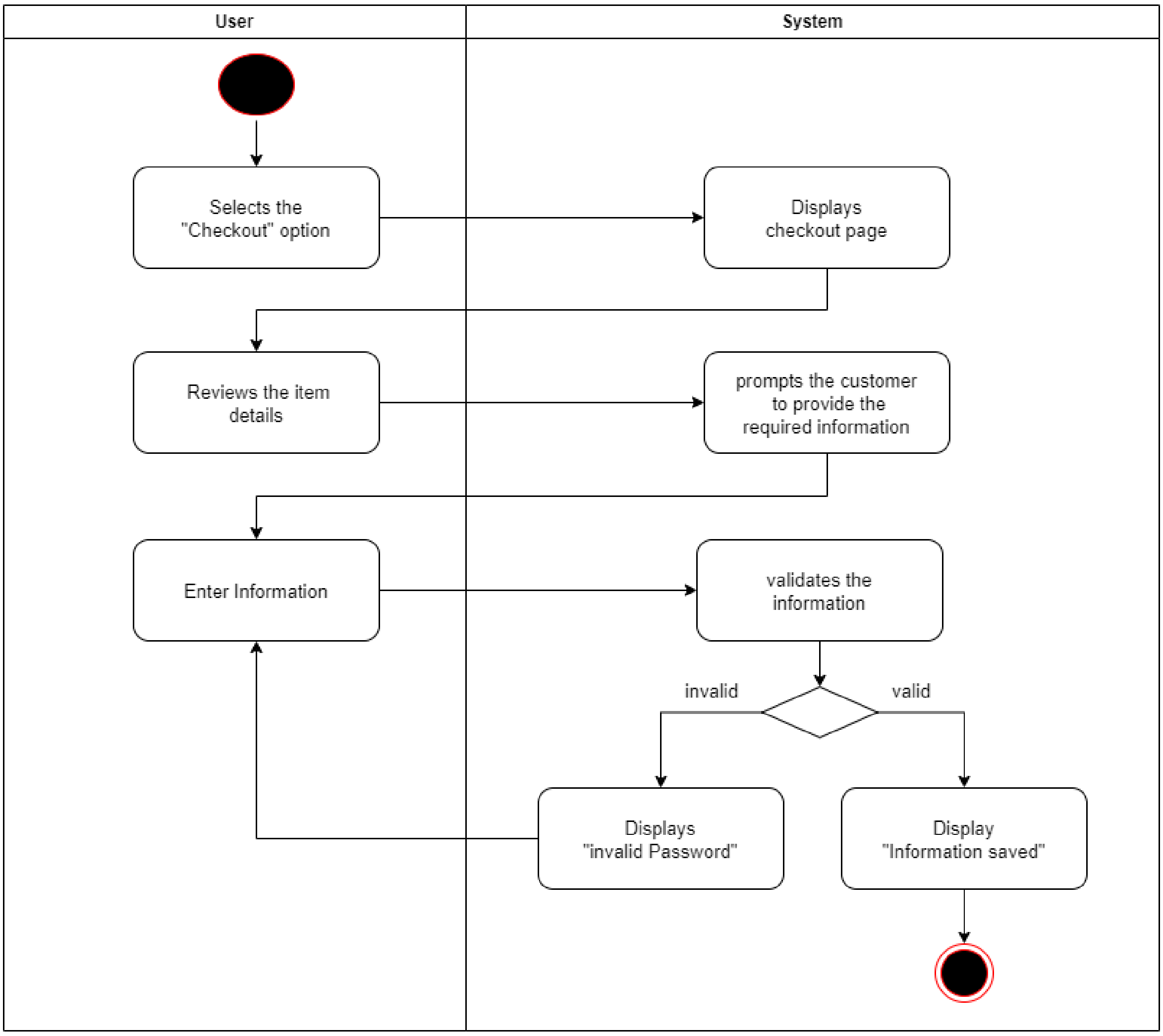
Update password Activity diagrams



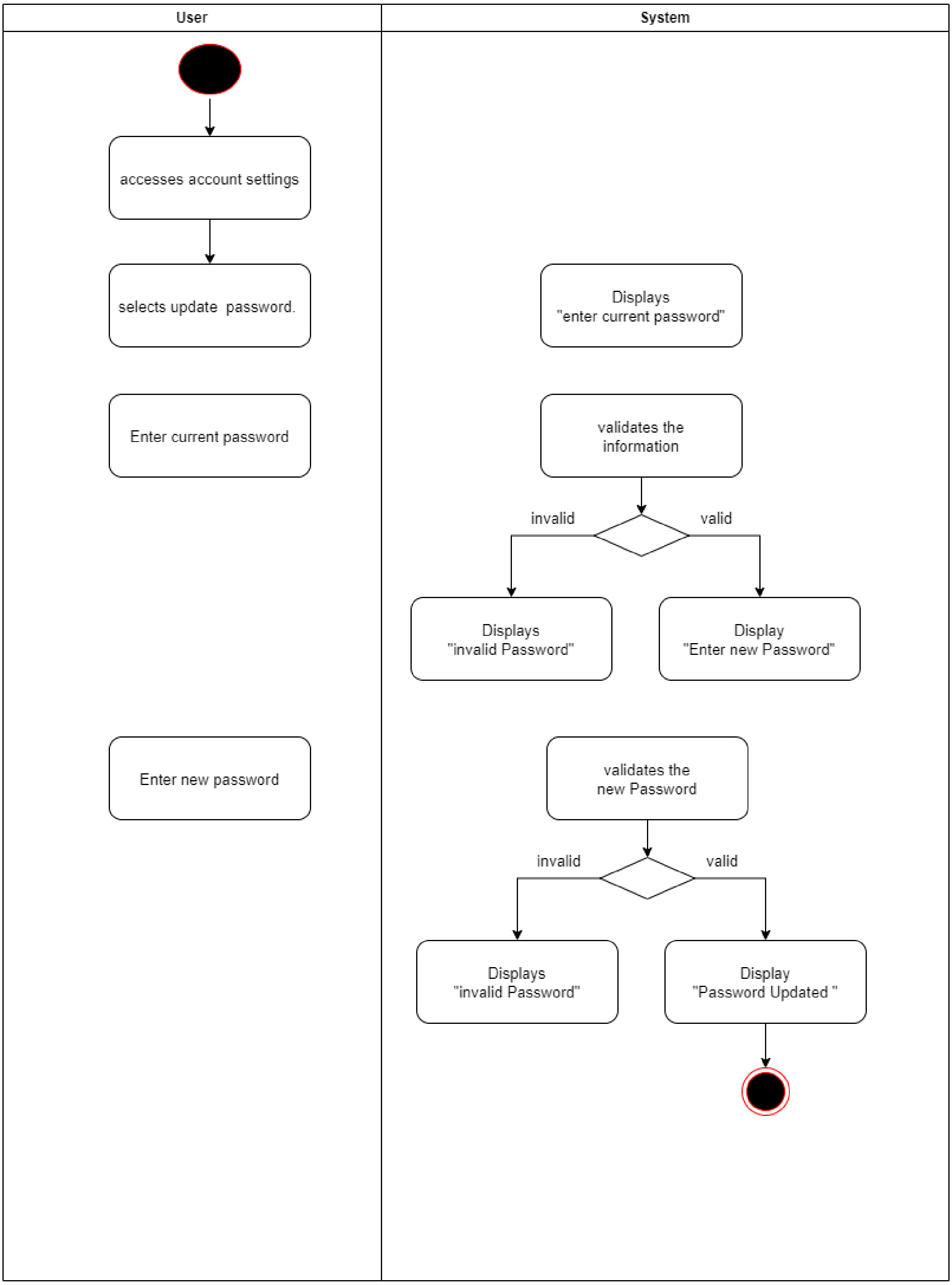
Update User profile Activity diagrams


User
System
accesses account settings
Update profile & submit
validates the information
invalid
valid
Displays "invalid information"
Display "successful profile update"

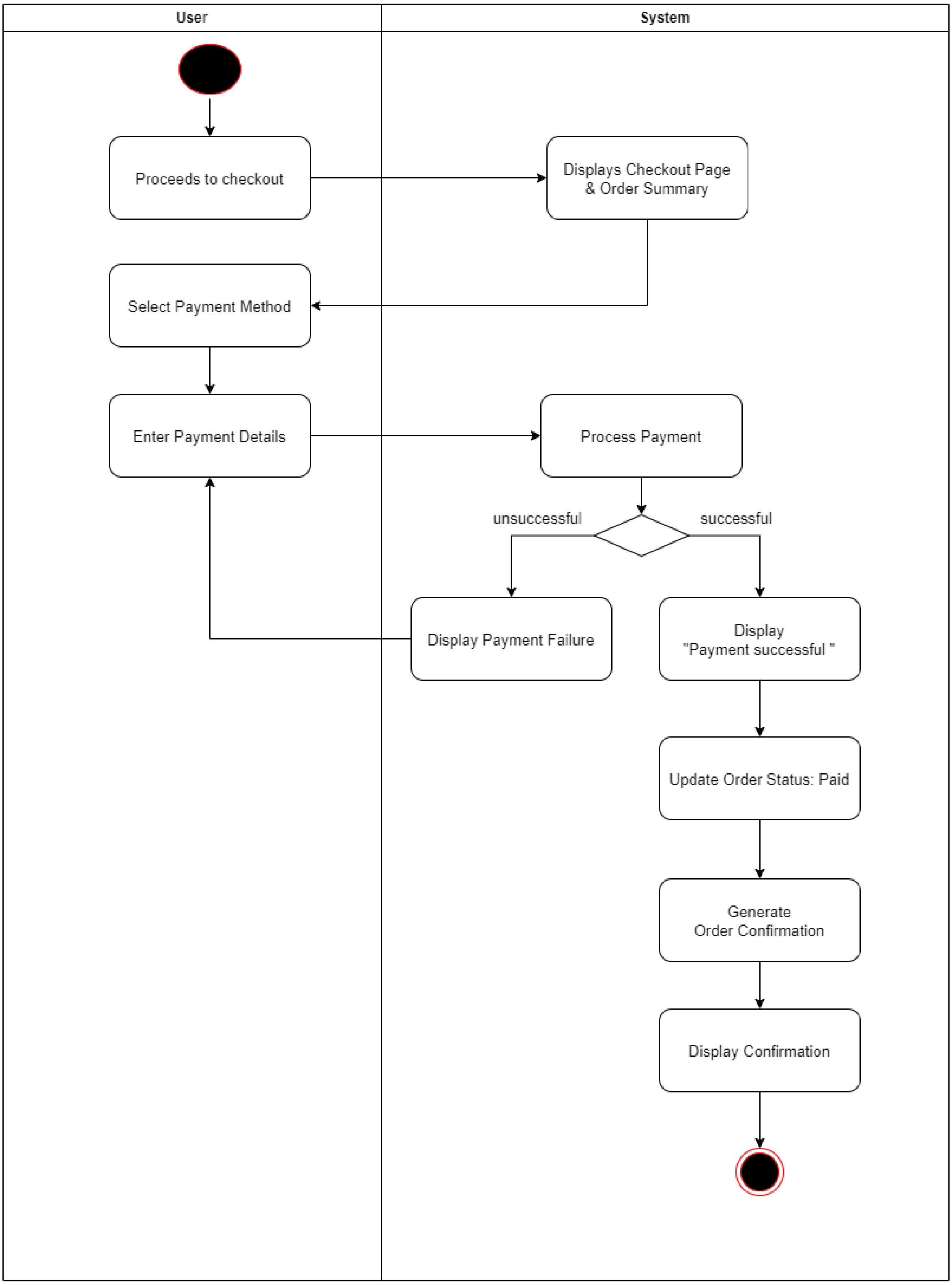
Check out Activity diagrams


User
System
Selects the "Checkout" option
Displays checkout page
Reviews the item details
prompts the customer to provide the required information
Enter Information
validates the information
invalid
valid
Displays "invalid Password"
Display "Information saved"

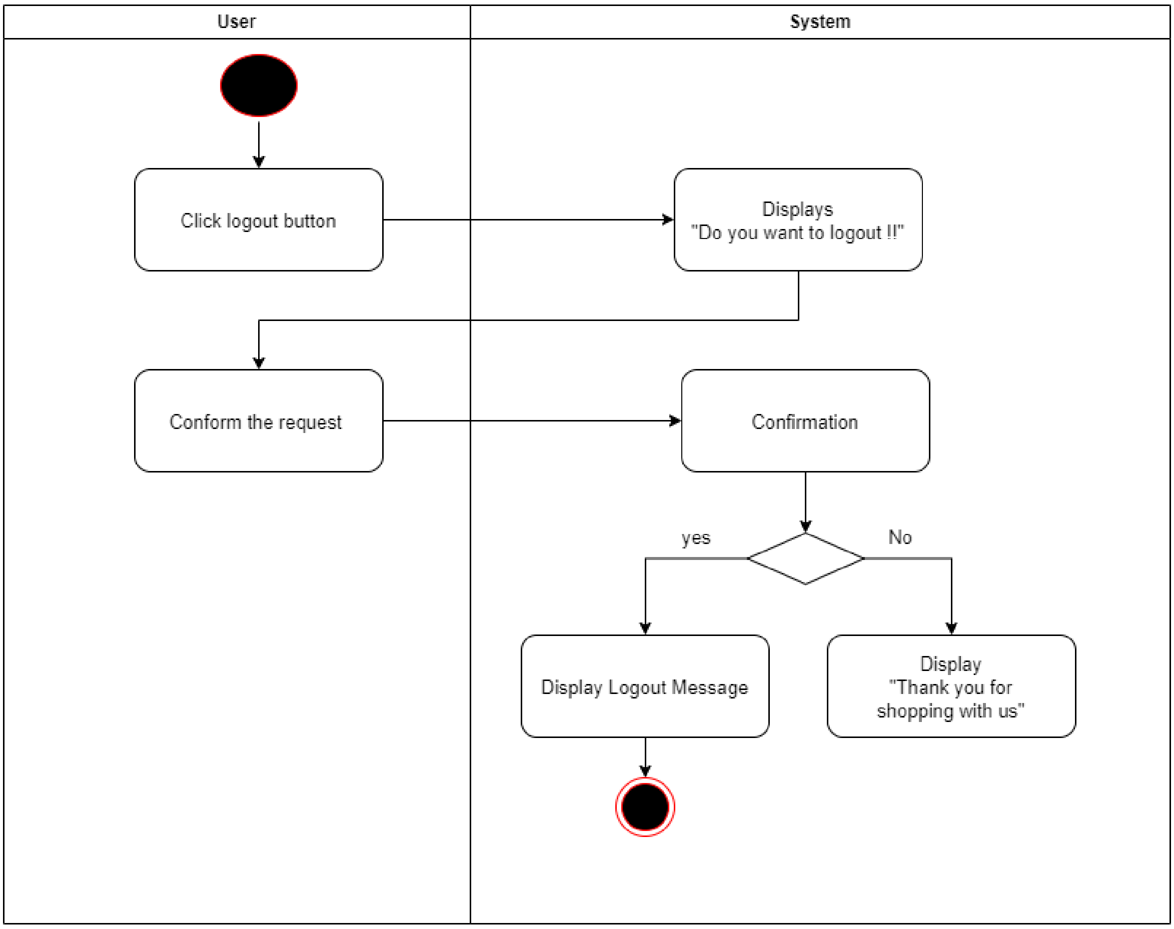
Add shipping information Activity diagrams


User
System
accesses account settings
selects update password.
Displays "enter current password"
Enter current password
validates the information
invalid
valid
Displays "invalid Password"
Display "Enter new Password"
Enter new password
validates the new Password
invalid
valid
Displays "invalid Password"
Display "Password Updated "

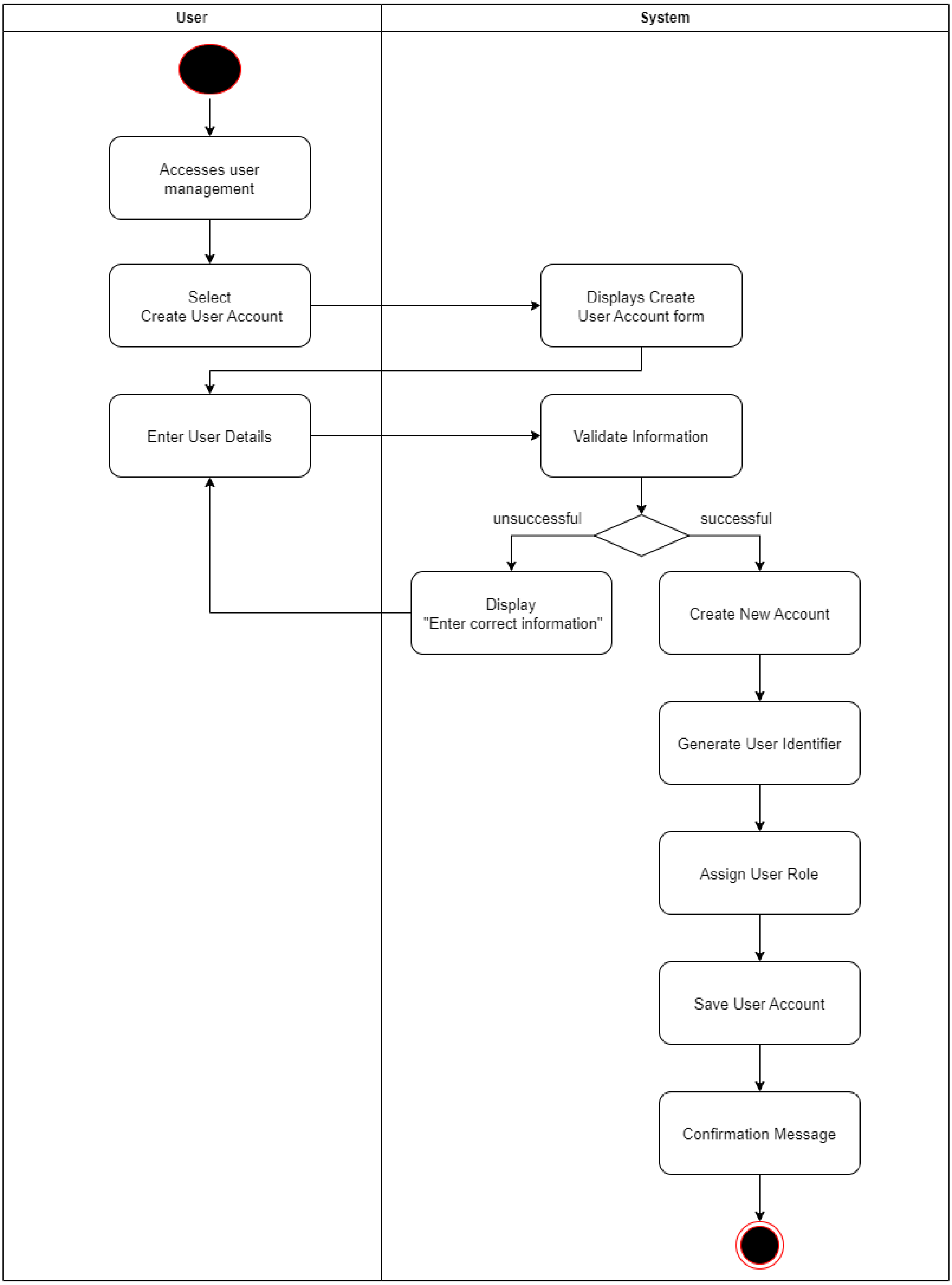
Add Payment Activity diagrams



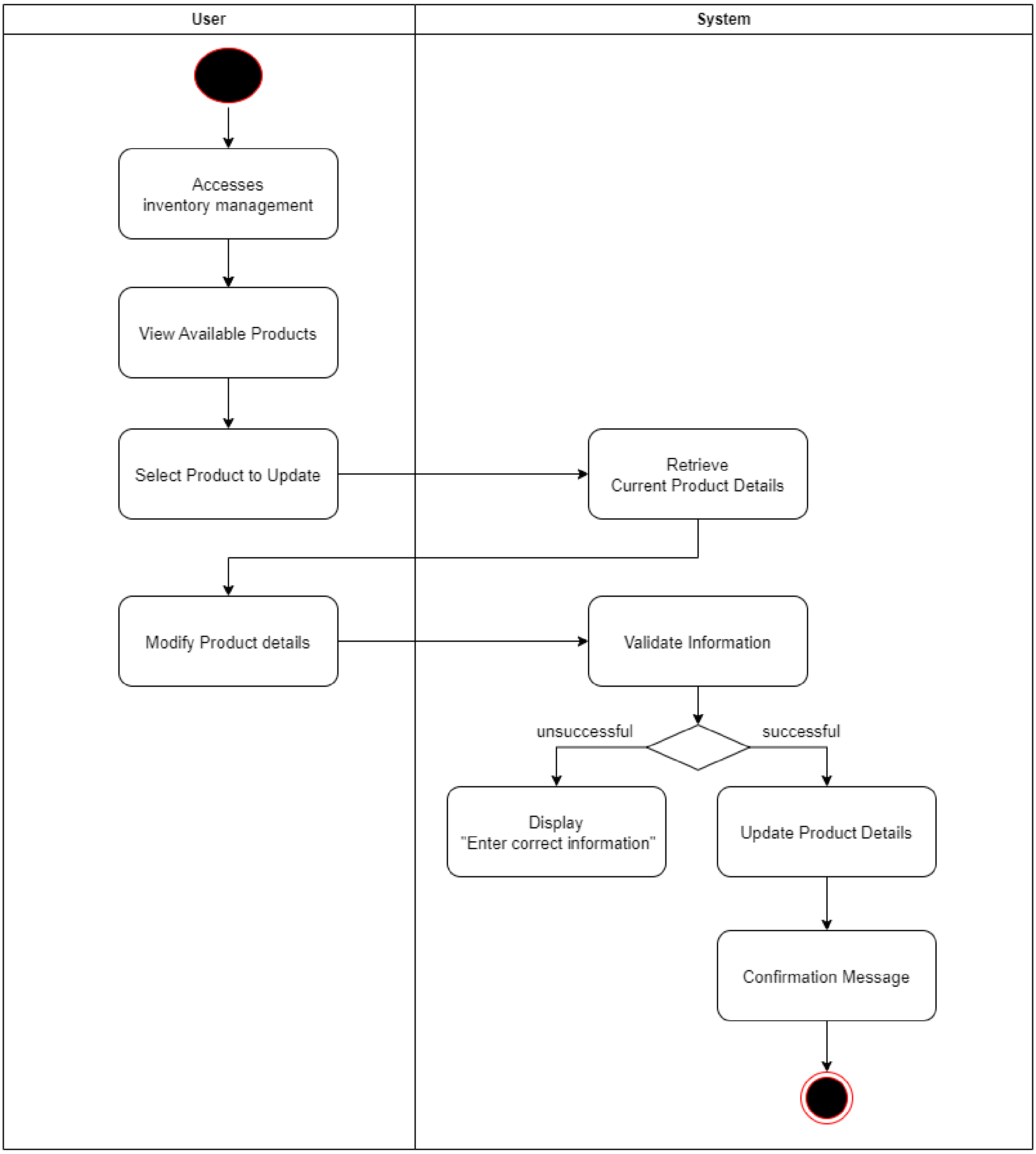
Logout Activity diagrams



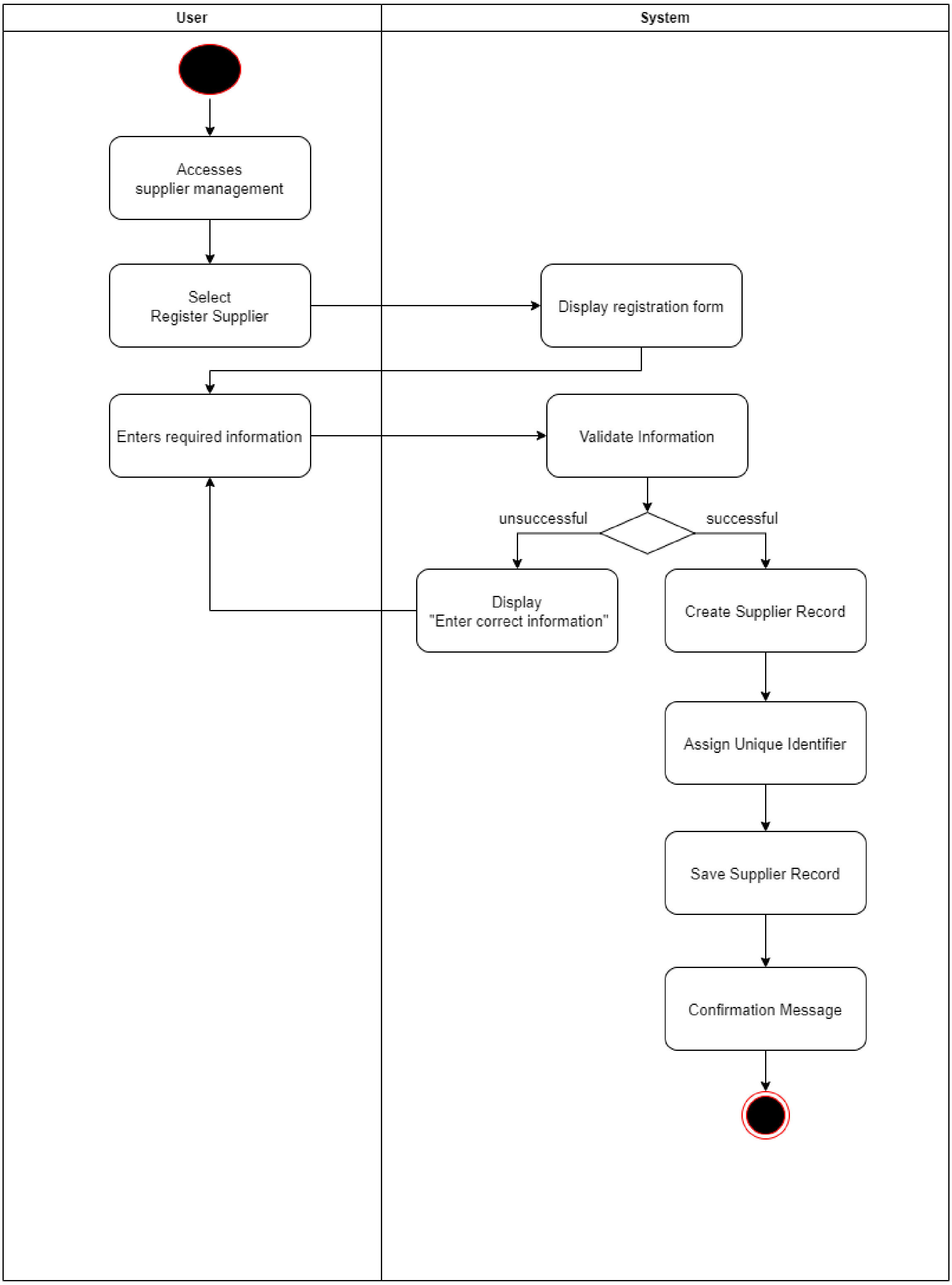
Create Management account Activity diagrams



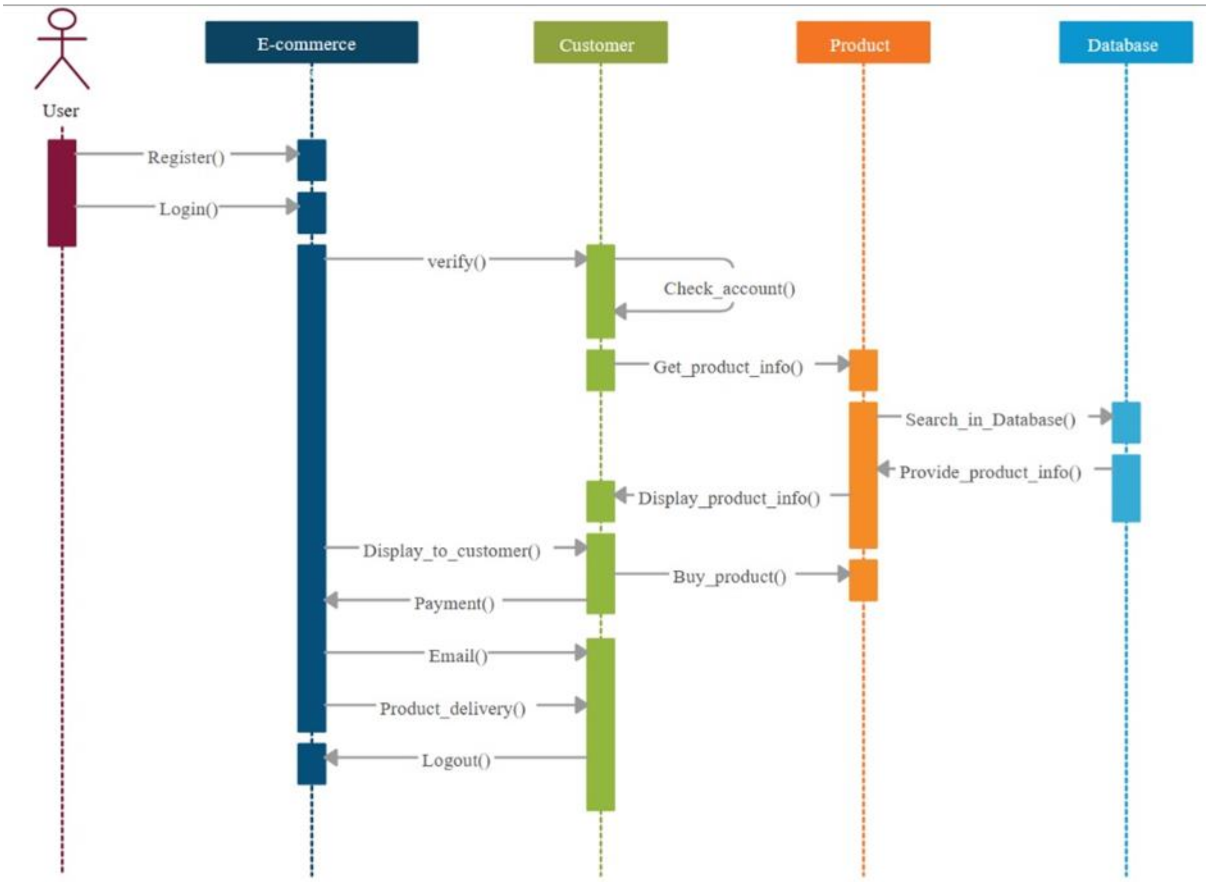
Product update Activity diagrams



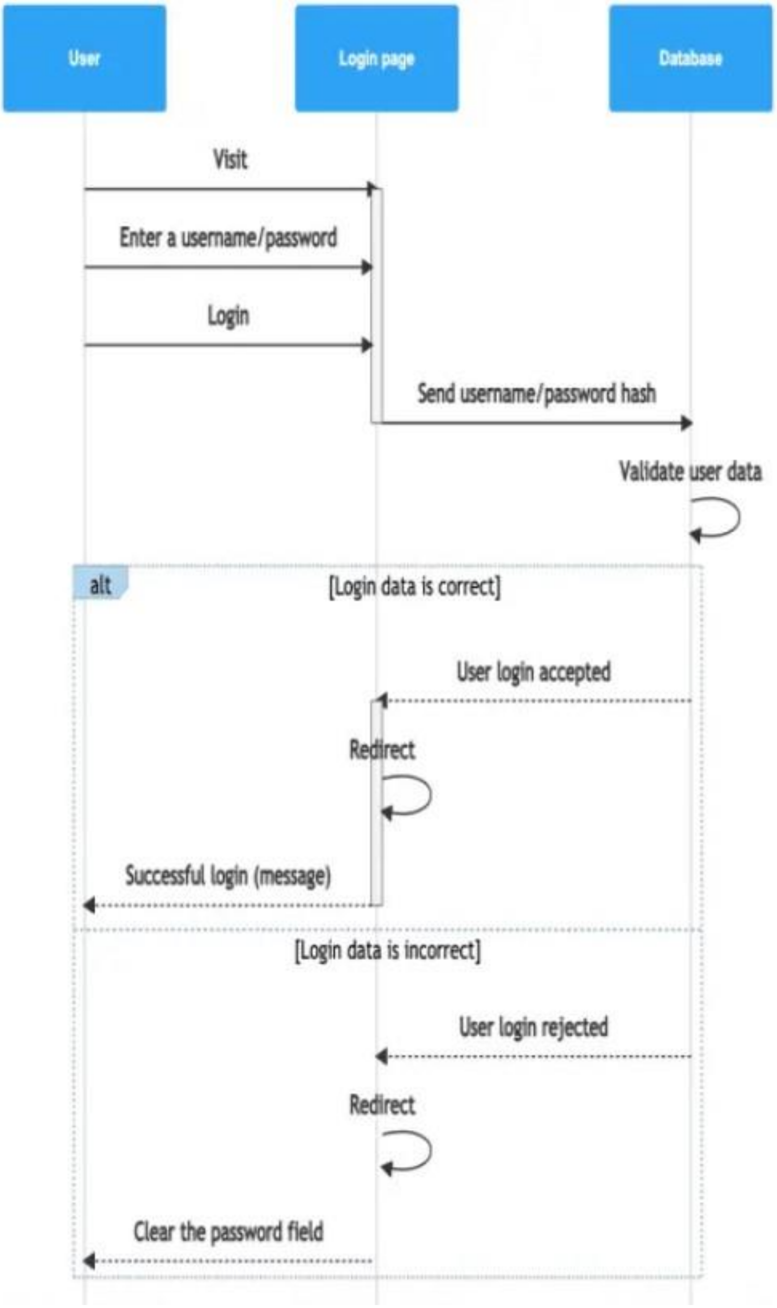
Supplier registration Activity diagrams



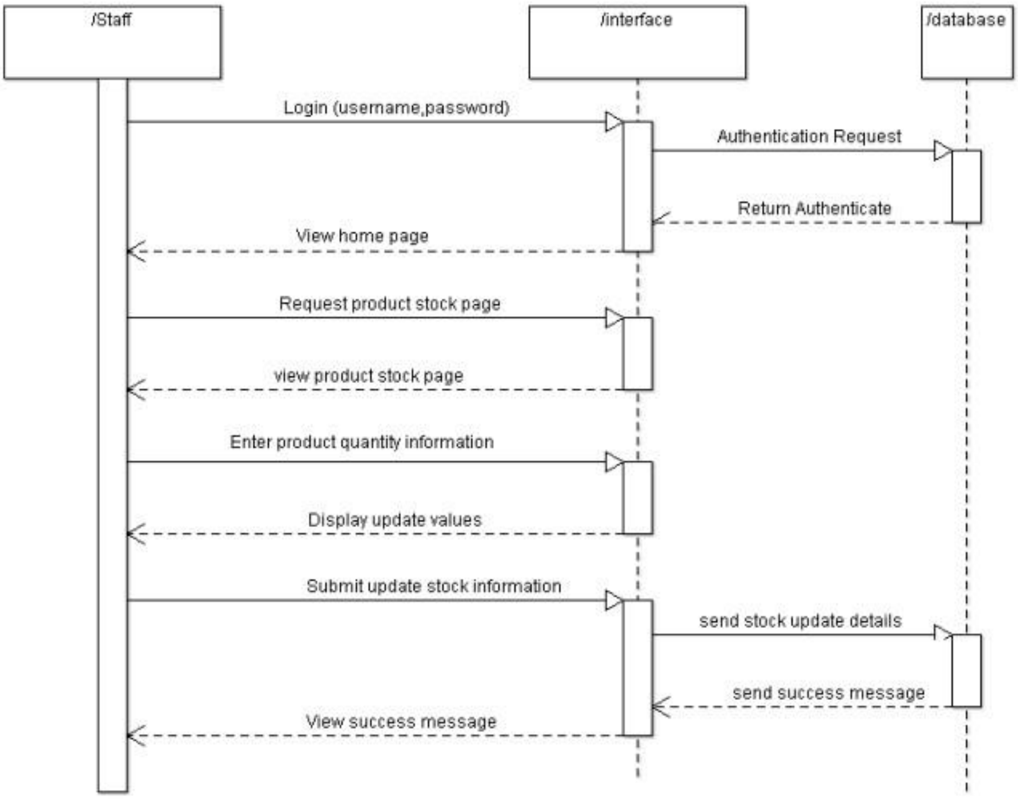
Shopping cart sequence diagram



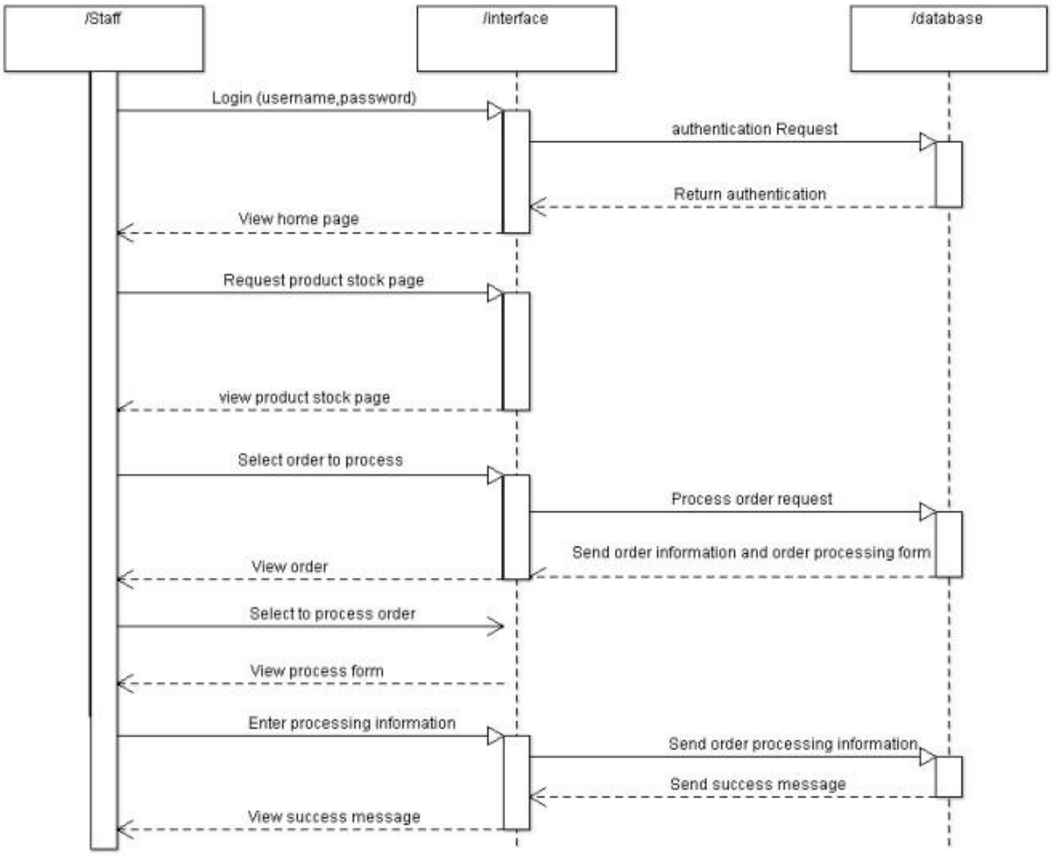
Login sequence diagram



Stock update sequence diagram



Order processing sequence diagram



5.3 Hardware and software requirements

Hardware requirements

Server

- Processor – Intel Core i7 fourth-generation or higher.
- Ram – 8GB
- Storage – 20GB HDD Drive
- 1Mbps network connection

End-User

- Processor – Intel Core i3 sixth-generation or higher.
- Ram – 4GB
- 1Mbps network connection

Software requirements

Server

- Windows 8 or higher OS
- Apache server
- PHP 7.4 higher
- MySql

End-User

- Windows 8 or higher OS
- Chromium-based v76 or Firefox v73 or higher web browser

5.4 Evaluating of solutions

In the design phase of the shopping cart and stock management web application, it is crucial to evaluate different solutions to ensure that they meet the requirements and goals of Future Tech. This evaluation process involves considering various factors such as feasibility, scalability, cost-effectiveness, and alignment with the business objectives. Here are the key aspects to consider when evaluating solutions:

1. Feasibility:
   - Technical Feasibility: Assess whether the proposed solution aligns with the existing technology infrastructure and can be implemented within the available resources and time frame.
   - Operational Feasibility: Evaluate whether the solution can be effectively integrated into the existing business processes and workflows, minimizing disruption and maximizing efficiency.

2. Scalability:
   - Consider the scalability of the solution to accommodate the future growth of Future Tech's customer base and product catalog. The solution should be able to handle increasing website traffic, order volume, and data processing requirements without significant performance degradation.

3. Cost-effectiveness:
   - Evaluate the cost implications of implementing and maintaining the solution. Consider factors such as development costs, licensing fees, infrastructure costs, and ongoing support and maintenance expenses. The chosen solution should provide value for money and align with Future Tech's budget constraints.

4. User Experience:
   - Assess the user experience offered by each solution. Consider factors such as ease of use, responsiveness, accessibility, and mobile responsiveness. The solution should provide a seamless and intuitive shopping experience for customers, promoting engagement and satisfaction.

5. Integration:
   - Evaluate the ability of the solution to integrate with other systems and services, such as payment gateways, shipping providers, and customer relationship management (CRM) tools. Seamless integration ensures smooth data flow and enhances operational efficiency.

6. Security:
   - Consider the security features and measures provided by each solution. Evaluate the ability to handle secure payment transactions, protect customer data, implement user authentication and authorization mechanisms, and comply with relevant data protection regulations.

7. Customization:
   - Assess the flexibility and customization options offered by each solution. Future Tech may have specific requirements and unique business processes that need to be accommodated. The solution should allow for customization and tailoring to meet these specific needs.

8. Vendor Reputation and Support:

- Evaluate the reputation and track record of the solution providers. Consider factors such as their experience in the industry, customer reviews and testimonials, and the availability of reliable technical support and maintenance services.

By thoroughly evaluating different solutions based on these factors, Future Tech can make an informed decision on the most suitable solution for their shopping cart and stock management web application. It is essential to prioritize solutions that align with the requirements and goals identified earlier and offer the best combination of functionality, usability, scalability, and cost-effectiveness.

Chapter 6 -Implementation

6.1 Introduction

The implementation phase is a critical stage in the development of the shopping cart and stock management web application for Future Tech. This chapter will outline the process and methodologies involved in implementing the solution, following the waterfall methodology. The implementation phase encompasses planning, analysis, design, actual implementation, testing, and maintenance. Each of these stages plays a crucial role in ensuring the successful development and deployment of the web application.

6.2 Implementation Methodology

The implementation methodology for the shopping cart and stock management web application follows the waterfall methodology. This sequential approach allows for a structured and systematic progression through the various stages of development.

6.2.1 Planning

The planning phase sets the foundation for the implementation process. It involves defining the project scope, objectives, and deliverables. Key tasks in this phase include:

- Defining the project timeline, milestones, and resource allocation.
- Establishing communication channels between the project owner and developer.
- Creating a detailed project plan, outlining the tasks, dependencies, and responsibilities.
- Identifying potential risks and developing risk mitigation strategies.

6.2.2 Analysis

The analysis phase involves gathering and documenting the detailed requirements for the shopping cart and stock management web application. Key tasks in this phase include:

- Conducting meetings and discussions with stakeholders to elicit and clarify requirements.
- Analyzing the existing business processes and identifying areas for improvement.
- Creating use cases, user stories, and functional requirements documentation.
- Validating the requirements with stakeholders to ensure accuracy and completeness.

6.2.3 Design

The design phase focuses on translating the requirements into a technical design for the web application. Key tasks in this phase include:

- Creating a high-level system architecture that defines the overall structure of the application.
- Designing the database schema and data models to support the required functionalities.
- Developing wireframes, prototypes, and user interface designs to visualize the application's look and feel.
- Defining the data flow, system components, and interfaces between different modules.

6.2.4 Implementation

The implementation phase involves the actual development of the shopping cart and stock management web application based on the design specifications. Key tasks in this phase include:

- Writing code and building the software components following industry best practices and coding standards.
- Integrating the different modules and functionalities of the application.
- Developing the necessary APIs, database interactions, and server-side logic.
- Conducting regular code reviews and quality assurance checks to ensure the code meets the defined requirements.

6.2.5 Testing

The testing phase is crucial to identify and fix any defects or issues in the developed application. Key tasks in this phase include:

- Conducting unit testing to verify the functionality and correctness of individual components.
- Performing integration testing to ensure the seamless interaction between different modules.
- Conducting system testing to validate the overall behavior and performance of the web application.
- Performing user acceptance testing (UAT) to involve stakeholders in testing the application and gathering feedback.

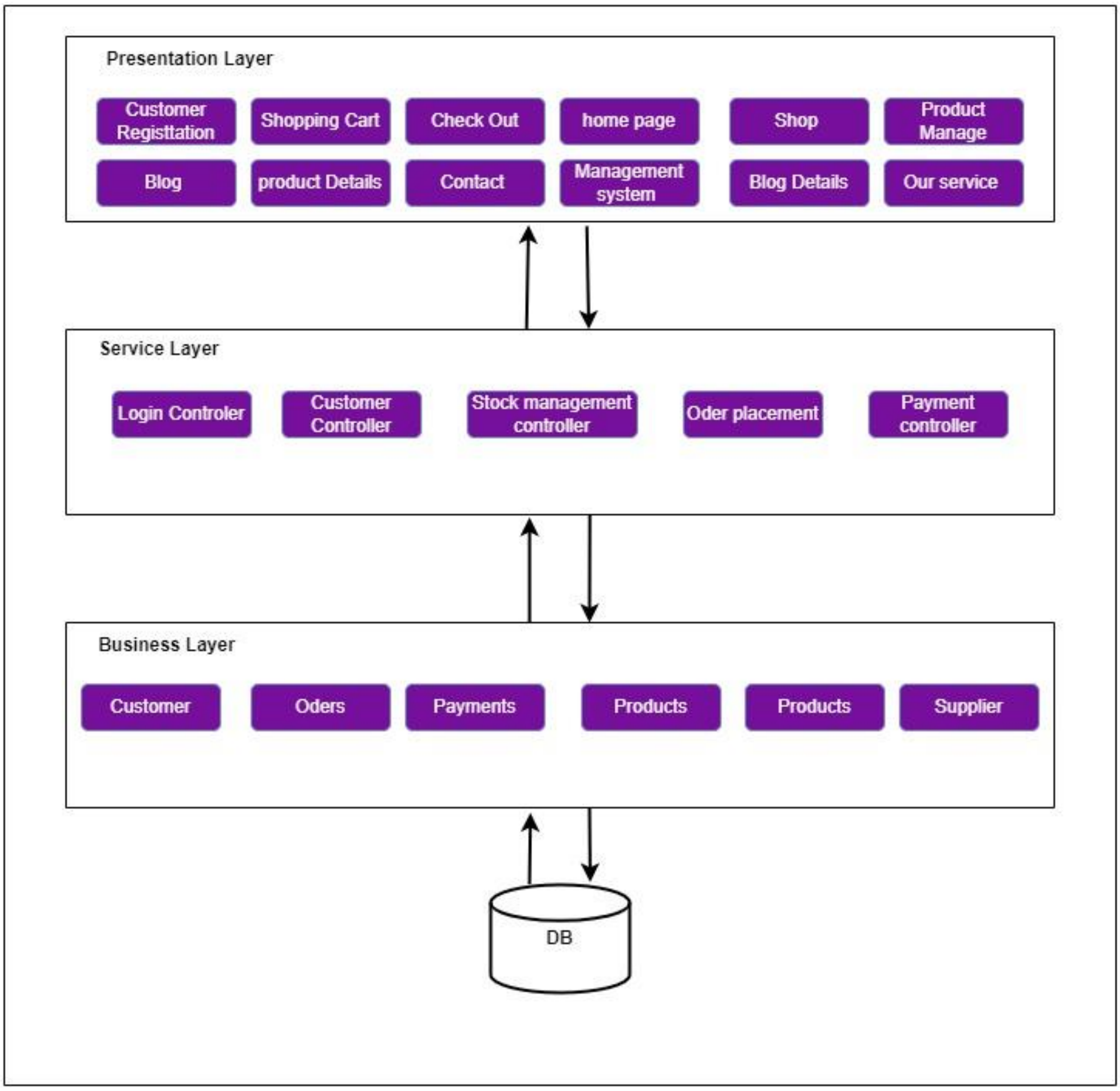
6.2.6 Maintenance

The maintenance phase involves deploying the web application to a production environment and providing ongoing support and enhancements. Key tasks in this phase include:

- Deploying the application to the production servers and ensuring its availability.
- Monitoring the application for performance, security, and stability.
- Addressing any reported issues or bugs through bug fixing and updates.
- Providing ongoing maintenance and support, including user training, documentation updates, and system enhancements based on feedback.

By following the waterfall methodology and executing each phase systematically, Future Tech can ensure a well-structured and organized implementation of the shopping cart and stock management web application. The implementation phase lays the groundwork for the subsequent stages of testing, deployment, and maintenance, ultimately leading to the successful launch of the web application.

6.3 System Architecture



6.4 Tools & Technologies used

During the implementation of the shopping cart and stock management web application, the following tools and technologies were utilized:

1. HTML (HyperText Markup Language):

- HTML is the standard markup language used for creating the structure and layout of web pages.

2. CSS (Cascading Style Sheets):

- CSS is used for styling and formatting the visual appearance of the web pages created with HTML.

3. PHP:

- PHP is a popular server-side scripting language used for developing dynamic web applications.
- PHP was used for implementing the business logic, handling server-side processing, and interacting with the database.

4. NetBeans:

- NetBeans is an integrated development environment (IDE) that provides tools for developing web applications.
- It supports various programming languages, including PHP, and offers features like code editing, debugging, and project management.

5. SQL (Structured Query Language):

- SQL is a language used for managing and manipulating databases.
- SQL was used for creating and querying the database for storing and retrieving data related to products, customers, orders, and other relevant information.

6. phpMyAdmin:

- phpMyAdmin is a web-based tool used for managing MySQL databases.
- It provides a user-friendly interface for performing administrative tasks, such as creating tables, executing queries, and managing database operations.

7. XAMPP:

- XAMPP is a cross-platform software package that includes Apache web server, MySQL database, PHP, and Perl.
- XAMPP was used as a local development environment for running and testing the web application on a personal computer.

These tools and technologies provided the necessary infrastructure and capabilities for developing the shopping cart and stock management web application. HTML and CSS were used for creating the user interface and defining the visual appearance of the web pages. PHP served as the server-side scripting language for implementing the application logic. NetBeans provided an integrated development environment for coding and debugging PHP files. SQL and phpMyAdmin were utilized for managing the database and executing SQL queries. XAMPP facilitated the local deployment and testing of the web application.

By leveraging these tools and technologies, Future Tech was able to build the necessary components and functionalities of the shopping cart and stock management web application, resulting in a functional and efficient system.

6.5 Database Implementation

The database implementation is a crucial aspect of the shopping cart and stock management web application. It involves designing and implementing the database schema to efficiently store and manage the application's data. Here is an overview of the steps involved in the database implementation:

1. Database Design:

- Define the entities, relationships, and attributes based on the requirements identified during the analysis phase.
- Create an entity-relationship diagram (ERD) to visualize the database structure and relationships between entities.
- Normalize the database design to eliminate data redundancy and ensure data integrity.

2. Database Management System (DBMS):

- Select a suitable DBMS that aligns with the project requirements and supports the desired features and scalability.
- Install and configure the chosen DBMS on the server or hosting environment.
- Set up authentication and access control mechanisms to ensure secure access to the database.

3. Table Creation:

- Create the necessary tables in the database based on the entities identified in the database design.
- Define the appropriate data types, constraints, and relationships (such as primary and foreign keys) for each table.
- Ensure that the table structure aligns with the normalized database design.

4. Data Insertion:

- Populate the database with initial data, such as product information, customer details, and other relevant data.
- Use SQL statements or database import tools to insert data into the respective tables.
- Validate the data for accuracy and consistency.

5. Indexing and Optimization:

- Identify the key fields and columns that will be frequently queried.
- Apply indexing techniques to improve query performance.
- Analyze and optimize database queries to ensure efficient data retrieval.

6. Data Integrity and Constraints:

- Define constraints, such as unique constraints, check constraints, and foreign key constraints, to enforce data integrity rules.
- Implement cascading actions, if required, to maintain referential integrity between related tables.
- Set up triggers or stored procedures to handle complex business rules or automated database operations.

7. Backup and Recovery:

- Implement regular database backup strategies to protect against data loss.
- Set up automated backup processes or use database backup tools to create regular backups of the database.
- Test the backup and recovery processes to ensure data can be restored successfully if needed.

8. Security Measures:

- Implement appropriate security measures to protect the database and data.
- Apply access controls and user privileges to restrict unauthorized access to sensitive data.
- Implement encryption techniques to secure sensitive data, such as customer information or payment details.

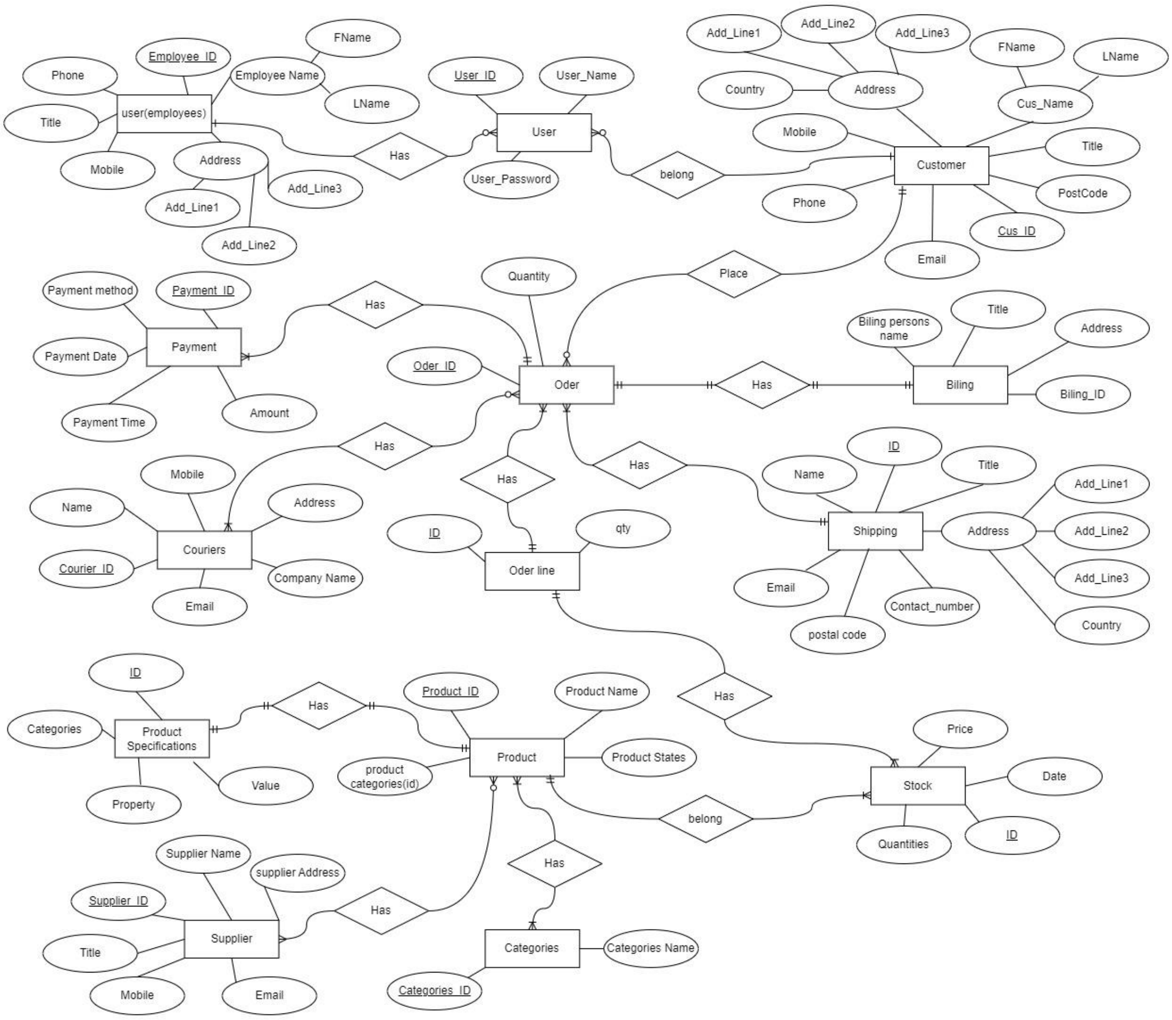
9. Testing and Verification:

- Perform data validation and verification to ensure that the data in the database is accurate and consistent.
- Test the database operations, such as data retrieval, insertion, modification, and deletion, to ensure they function as expected.
- Conduct performance testing to evaluate the database's responsiveness and scalability under different load conditions.

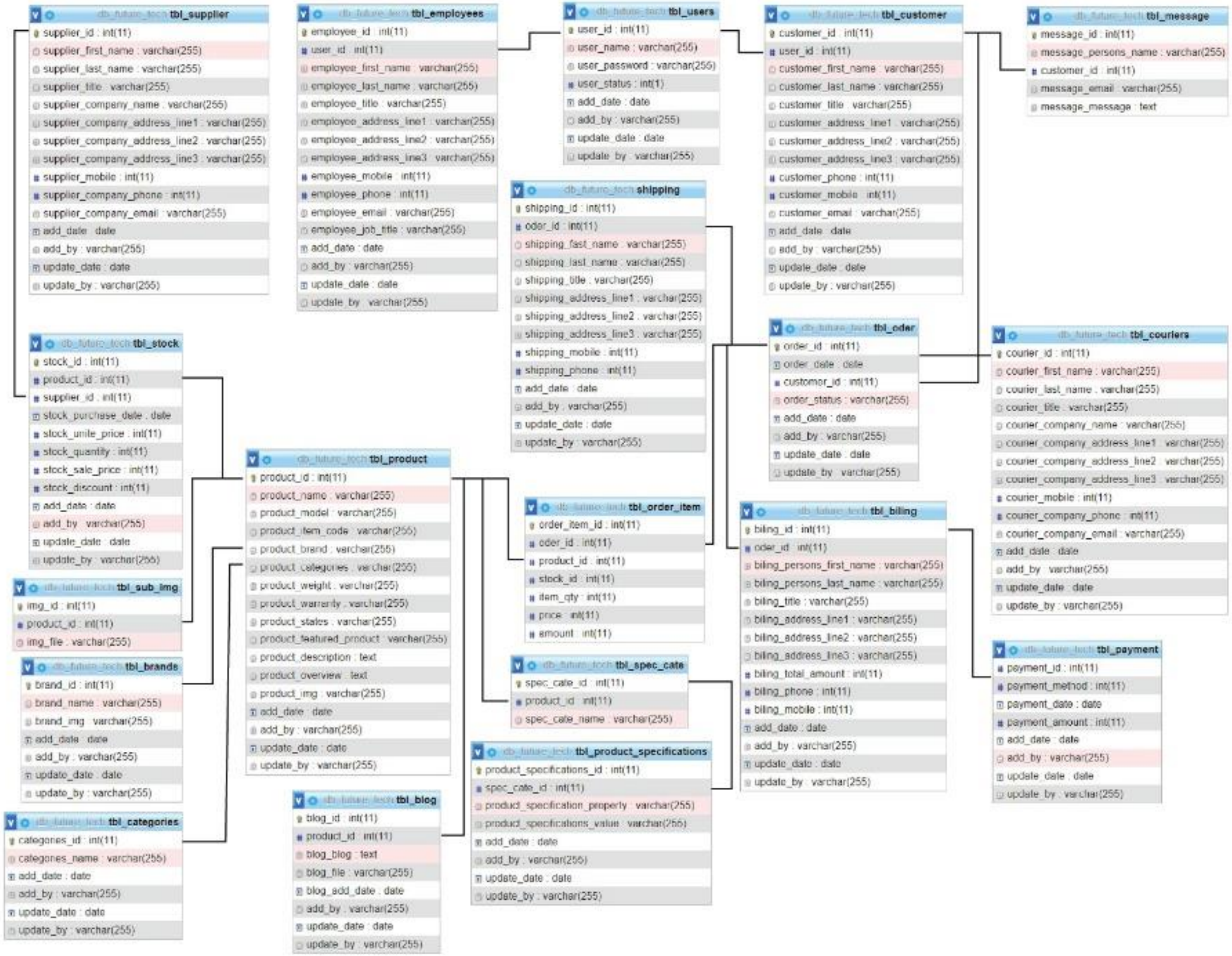
By following these steps, Future Tech can successfully implement the database for the shopping cart and stock management web application. A well-designed and efficiently implemented database is essential for the application's performance, data integrity, and overall functionality.

6.5.1 Internal Database Schema

ER Diagram:

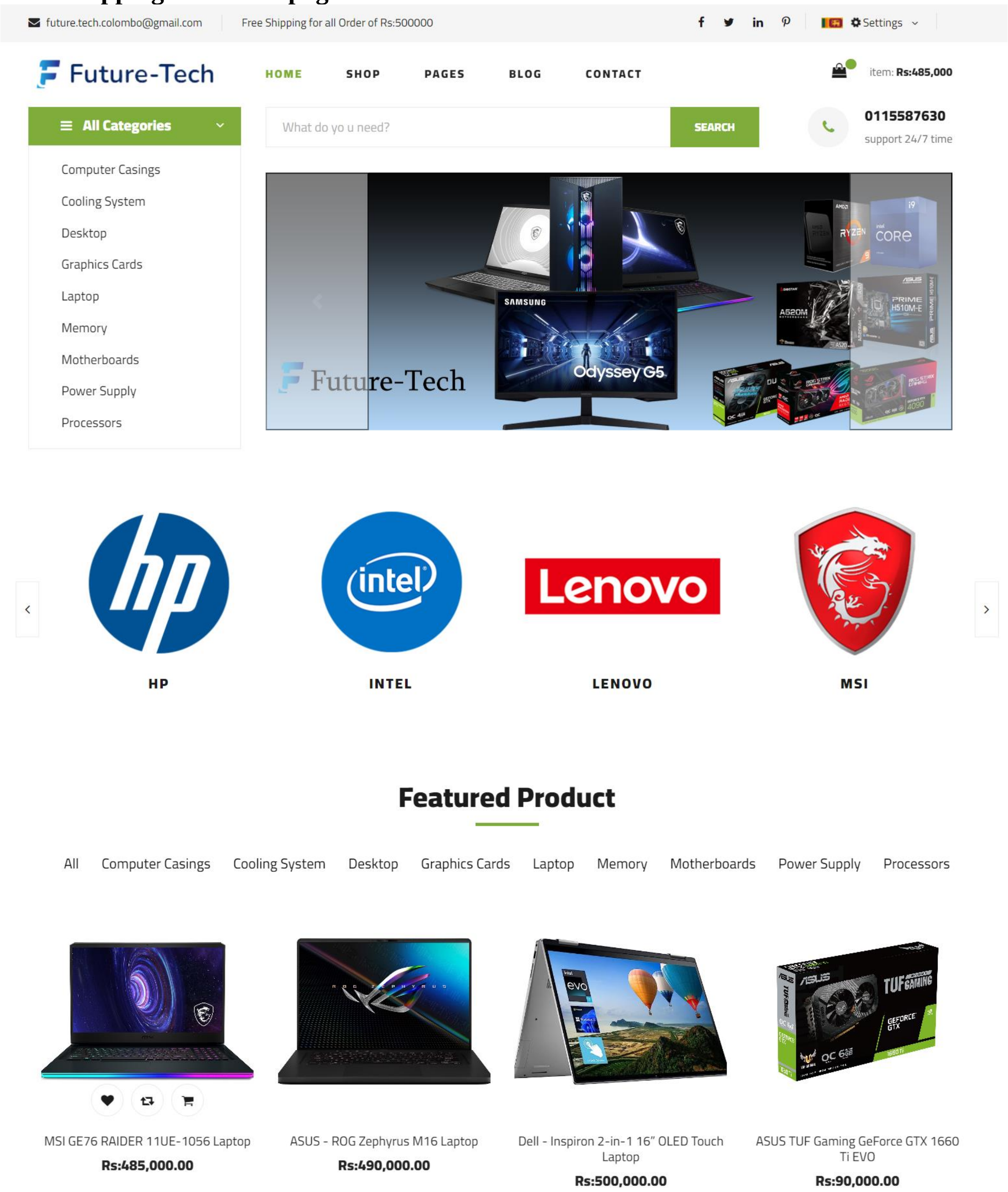


Data base diagram


db_future_tech tbl_supplier
supplier_id : int(11)
supplier_first_name : varchar(255)
supplier_last_name : varchar(255)
supplier_title : varchar(255)
supplier_company_name : varchar(255)
supplier_company_address_line1 : varchar(255)
supplier_company_address_line2 : varchar(255)
supplier_company_address_line3 : varchar(255)
supplier_mobile : int(11)
supplier_company_phone : int(11)
supplier_company_email : varchar(255)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_employees
employee_id : int(11)
user_id : int(11)
employee_first_name : varchar(255)
employee_last_name : varchar(255)
employee_title : varchar(255)
employee_address_line1 : varchar(255)
employee_address_line2 : varchar(255)
employee_address_line3 : varchar(255)
employee_mobile : int(11)
employee_phone : int(11)
employee_email : varchar(255)
employee_job_title : varchar(255)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_users
user_id : int(11)
user_name : varchar(255)
user_password : varchar(255)
user_status : int(1)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_customer
customer_id : int(11)
user_id : int(11)
customer_first_name : varchar(255)
customer_last_name : varchar(255)
customer_title : varchar(255)
customer_address_line1 : varchar(255)
customer_address_line2 : varchar(255)
customer_address_line3 : varchar(255)
customer_phone : int(11)
customer_mobile : int(11)
customer_email : varchar(255)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_message
message_id : int(11)
message_persons_name : varchar(255)
customer_id : int(11)
message_email : varchar(255)
message_message : text
db_future_tech shipping
shipping_id : int(11)
oder_id : int(11)
shipping_fast_name : varchar(255)
shipping_last_name : varchar(255)
shipping_title : varchar(255)
shipping_address_line1 : varchar(255)
shipping_address_line2 : varchar(255)
shipping_address_line3 : varchar(255)
shipping_mobile : int(11)
shipping_phone : int(11)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_oder
order_id : int(11)
order_date : date
customer_id : int(11)
order_status : varchar(255)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_couriers
courier_id : int(11)
courier_first_name : varchar(255)
courier_last_name : varchar(255)
courier_title : varchar(255)
courier_company_name : varchar(255)
courier_company_address_line1 : varchar(255)
courier_company_address_line2 : varchar(255)
courier_company_address_line3 : varchar(255)
courier_mobile : int(11)
courier_company_phone : int(11)
courier_company_email : varchar(255)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_stock
stock_id : int(11)
product_id : int(11)
supplier_id : int(11)
stock_purchase_date : date
stock_unite_price : int(11)
stock_quantity : int(11)
stock_sale_price : int(11)
stock_discount : int(11)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_product
product_id : int(11)
product_name : varchar(255)
product_model : varchar(255)
product_item_code : varchar(255)
product_brand : varchar(255)
product_categories : varchar(255)
product_weight : varchar(255)
product_warrenty : varchar(255)
product_states : varchar(255)
product_featured_product : varchar(255)
product_description : text
product_overview : text
product_img : varchar(255)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_order_item
order_item_id : int(11)
oder_id : int(11)
product_id : int(11)
stock_id : int(11)
item_qty : int(11)
price : int(11)
amount : int(11)
db_future_tech tbl_billing
biling_id : int(11)
oder_id : int(11)
biling_persons_first_name : varchar(255)
biling_persons_last_name : varchar(255)
biling_title : varchar(255)
biling_address_line1 : varchar(255)
biling_address_line2 : varchar(255)
biling_address_line3 : varchar(255)
biling_total_amount : int(11)
biling_phone : int(11)
biling_mobile : int(11)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_payment
payment_id : int(11)
payment_method : int(11)
payment_date : date
payment_amount : int(11)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_sub_img
img_id : int(11)
product_id : int(11)
img_file : varchar(255)
db_future_tech tbl_brands
brand_id : int(11)
brand_name : varchar(255)
brand_img : varchar(255)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_spec_cate
spec_cate_id : int(11)
product_id : int(11)
spec_cate_name : varchar(255)
db_future_tech tbl_product_specifications
product_specifications_id : int(11)
spec_cate_id : int(11)
product_specification_property : varchar(255)
product_specifications_value : varchar(255)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_categories
categories_id : int(11)
categories_name : varchar(255)
add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)
db_future_tech tbl_blog
blog_id : int(11)
product_id : int(11)
blog_blog : text
blog_file : varchar(255)
blog_add_date : date
add_by : varchar(255)
update_date : date
update_by : varchar(255)

6.6 User Interface Design

6.6.1 Shopping cart home page



future.tech.colombo@gmail.com Free Shipping for all Order of Rs:500000 Settings

Future-Tech HOME SHOP PAGES BLOG CONTACT item: Rs:485,000

All Categories What do yo u need? SEARCH 0115587630 support 24/7 time

- Computer Casings
- Cooling System
- Desktop
- Graphics Cards
- Laptop
- Memory
- Motherboards
- Power Supply
- Processors

HP INTEL LENOVO MSI

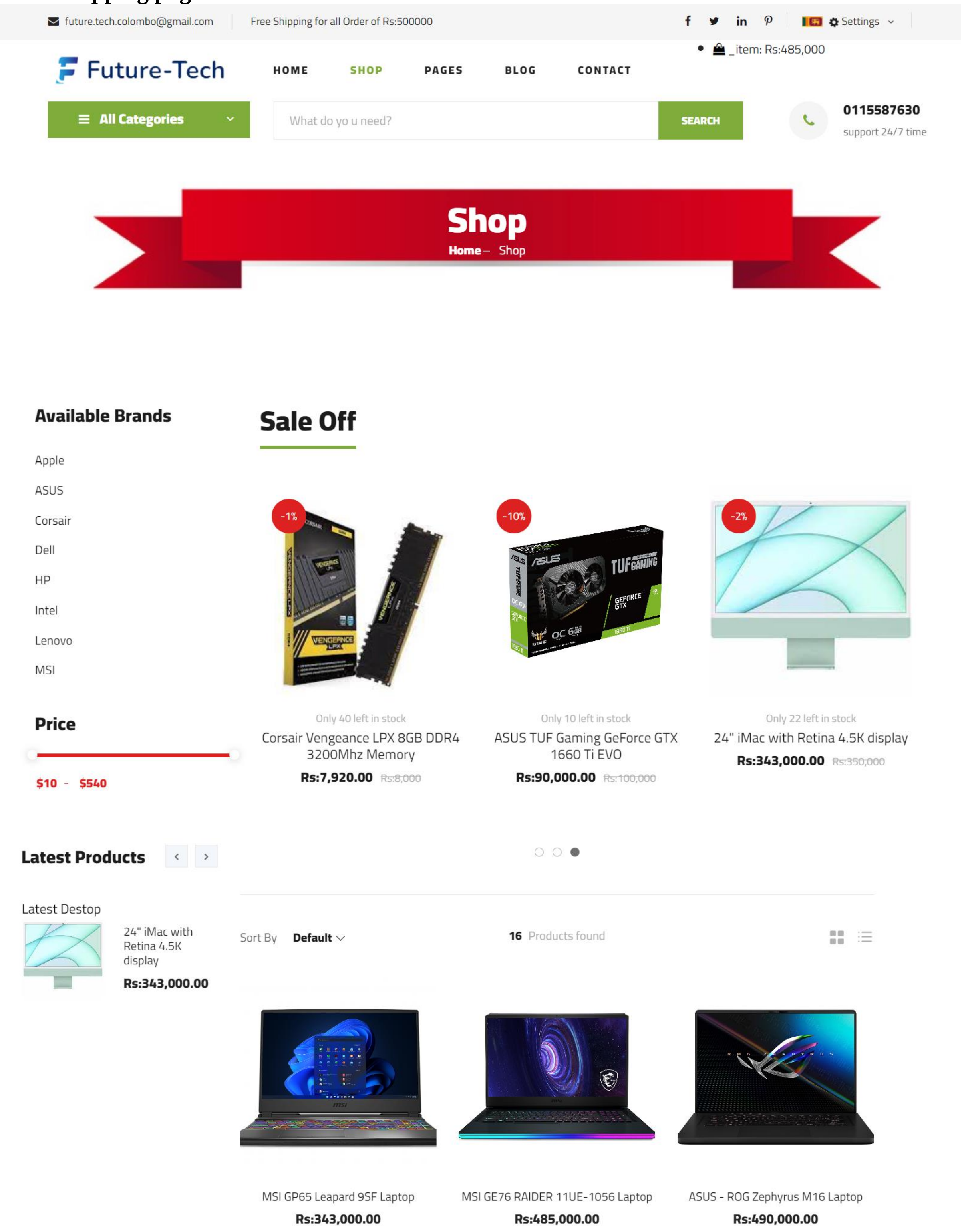
Featured Product

All Computer Casings Cooling System Desktop Graphics Cards Laptop Memory Motherboards Power Supply Processors

MSI GE76 RAIDER 11UE-1056 Laptop	ASUS - ROG Zephyrus M16 Laptop	Dell - Inspiron 2-in-1 16" OLED Touch Laptop	ASUS TUF Gaming GeForce GTX 1660 Ti EVO
Rs:485,000.00	Rs:490,000.00	Rs:500,000.00	Rs:90,000.00

The Shopping Cart Home Page serves as the primary entry point for users to access their shopping cart and manage their selected items. It provides a clear and user-friendly interface to enhance the shopping experience. Overall, the Shopping Cart Home Page is designed with simplicity and ease of use in mind, ensuring a seamless shopping experience for customers. The layout and intuitive features aim to reduce friction during the purchasing process and encourage users to complete their transactions.

6.6.2 Shopping page



future.tech.colombo@gmail.com | Free Shipping for all Order of Rs:500000

Settings

- _item: Rs:485,000

Future-Tech

HOME SHOP PAGES BLOG CONTACT

All Categories

What do yo u need? SEARCH

0115587630
support 24/7 time

Shop
Home– Shop

Available Brands

Apple
ASUS
Corsair
Dell
HP
Intel
Lenovo
MSI

Price

$10 - $540

Sale Off

-1%
Only 40 left in stock
Corsair Vengeance LPX 8GB DDR4 3200Mhz Memory
Rs:7,920.00 ~~Rs:8,000~~

-10%
Only 10 left in stock
ASUS TUF Gaming GeForce GTX 1660 Ti EVO
Rs:90,000.00 ~~Rs:100,000~~

-2%
Only 22 left in stock
24" iMac with Retina 4.5K display
Rs:343,000.00 ~~Rs:350,000~~

Latest Products

Latest Destop
24" iMac with Retina 4.5K display
Rs:343,000.00

Sort By Default

16 Products found

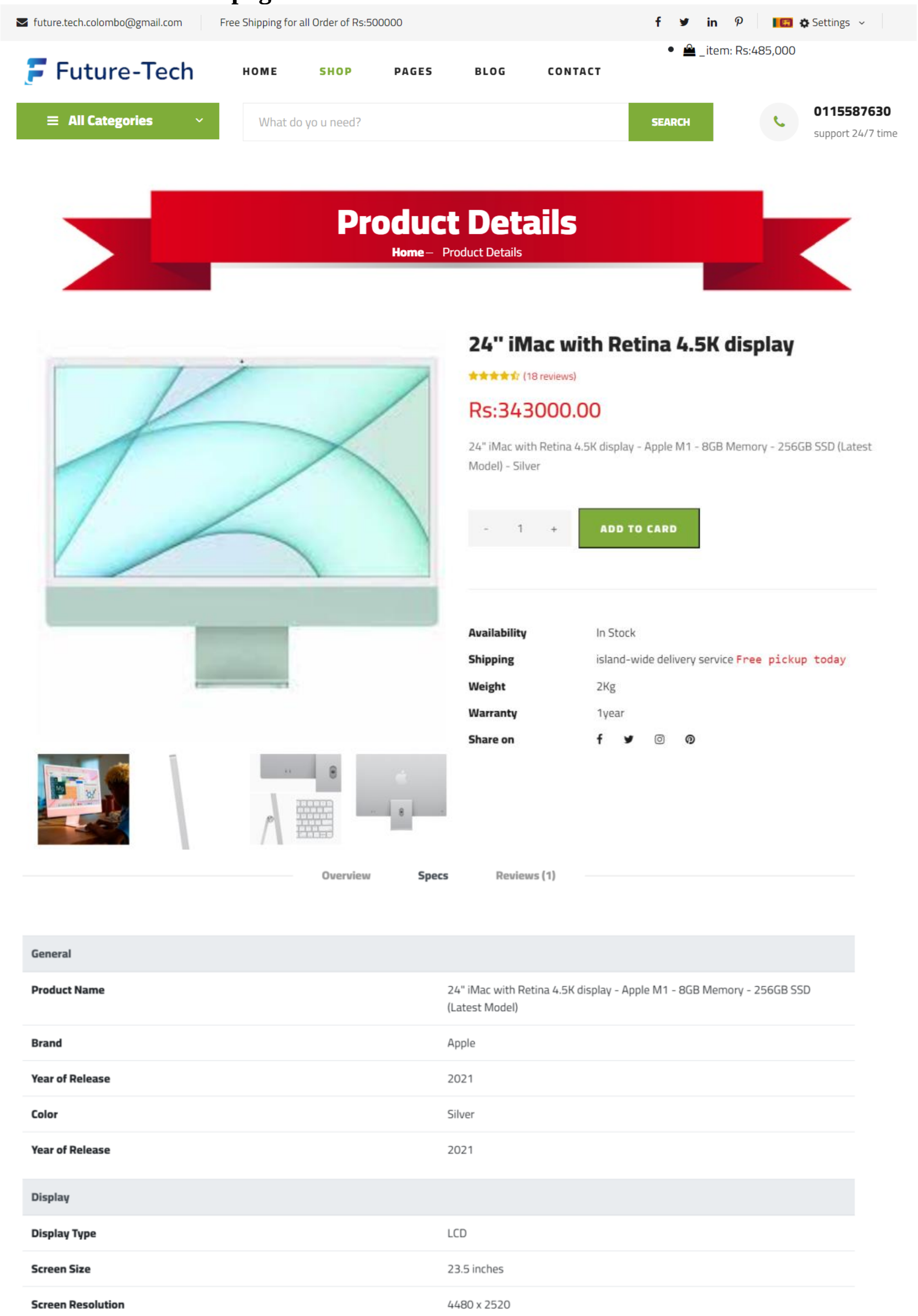
MSI GP65 Leapard 9SF Laptop
Rs:343,000.00

MSI GE76 RAIDER 11UE-1056 Laptop
Rs:485,000.00

ASUS - ROG Zephyrus M16 Laptop
Rs:490,000.00

The Shopping Page is the main catalog of products available in the online store. It is designed to showcase various products in an organized and visually appealing manner. The Shopping Page aims to provide a delightful shopping experience by presenting a wide range of products in an organized and visually appealing manner. The combination of product filters, sorting options, and detailed product information empowers users to make informed decisions and easily find the items they desire.

6.6.3 Product details page



24" iMac with Retina 4.5K display

★★★★★ (18 reviews)

Rs:343000.00

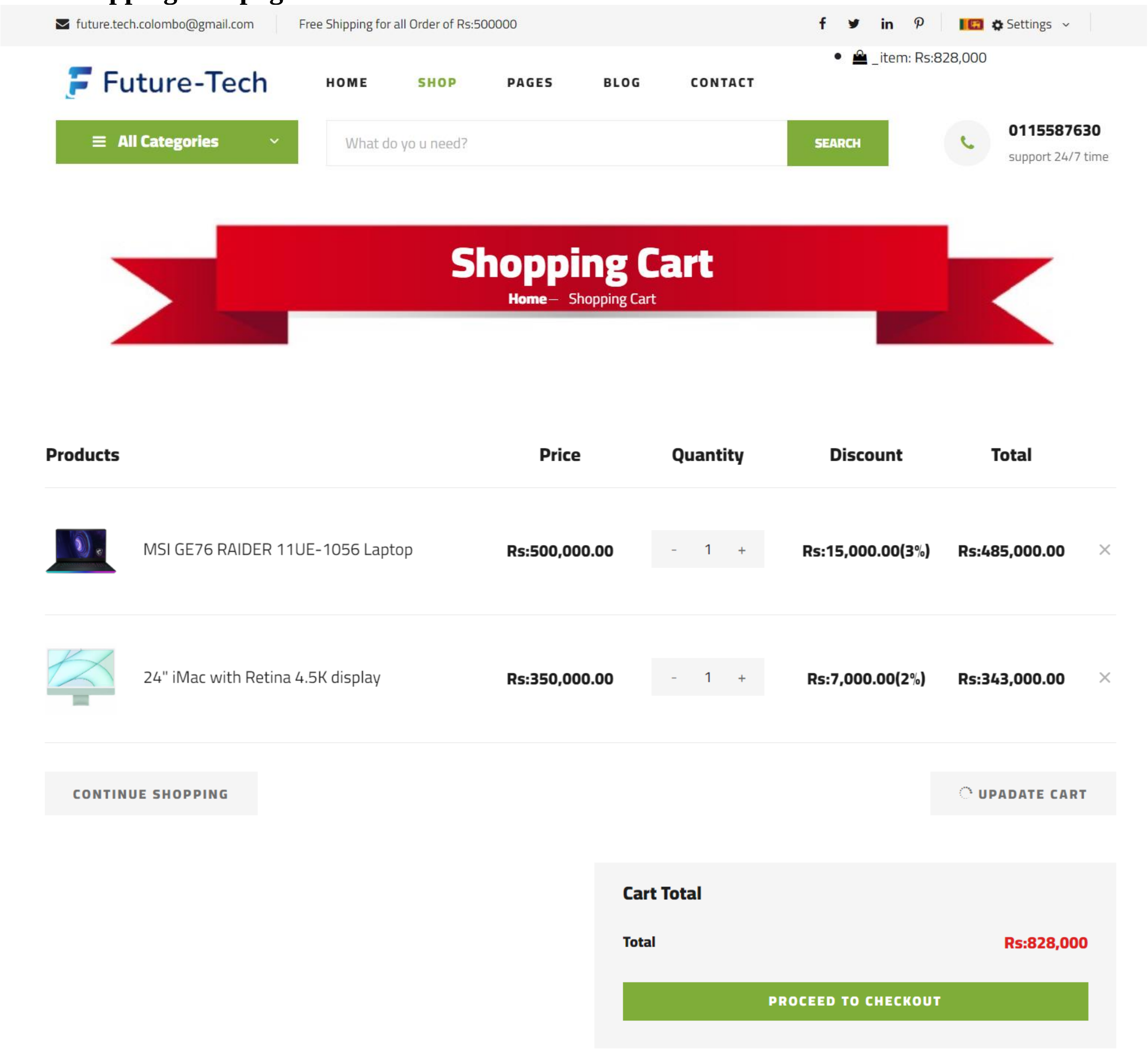
24" iMac with Retina 4.5K display - Apple M1 - 8GB Memory - 256GB SSD (Latest Model) - Silver

Availability	In Stock
Shipping	island-wide delivery service Free pickup today
Weight	2Kg
Warranty	1year
Share on	

General	
Product Name	24" iMac with Retina 4.5K display - Apple M1 - 8GB Memory - 256GB SSD (Latest Model)
Brand	Apple
Year of Release	2021
Color	Silver
Year of Release	2021
Display	
Display Type	LCD
Screen Size	23.5 inches
Screen Resolution	4480 x 2520

The Product Details Page is a dedicated page that provides comprehensive information about a specific product in the online store. This page is accessed when a user clicks on a product thumbnail or name from the Shopping Page. The Product Details Page is designed to give users a closer look at the chosen product and facilitate their decision-making process.

6.6.4 Shopping cart page



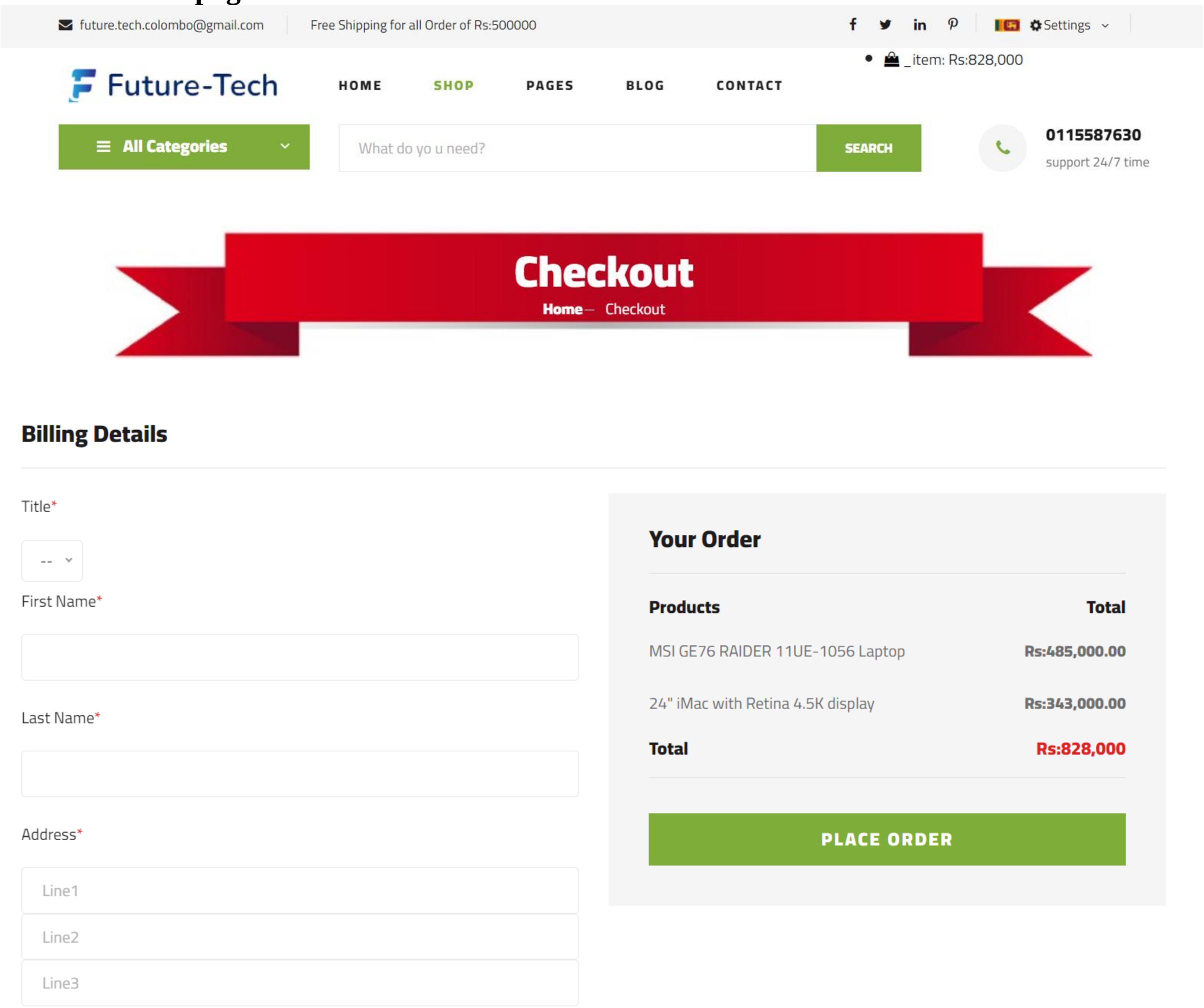
Products	Price	Quantity	Discount	Total
MSI GE76 RAIDER 11UE-1056 Laptop	Rs:500,000.00	- 1 +	Rs:15,000.00(3%)	Rs:485,000.00
24" iMac with Retina 4.5K display	Rs:350,000.00	- 1 +	Rs:7,000.00(2%)	Rs:343,000.00

Cart Total

Total	Rs:828,000

The Shopping Cart Page is a crucial part of the online store's user interface, providing users with an overview of the items they have added to their cart during their shopping session. The Shopping Cart Page is accessible through the shopping cart icon or a dedicated "View Cart" button located on various pages of the website. The primary purpose of this page is to display the selected products, allow users to review their choices, and proceed to the checkout process.

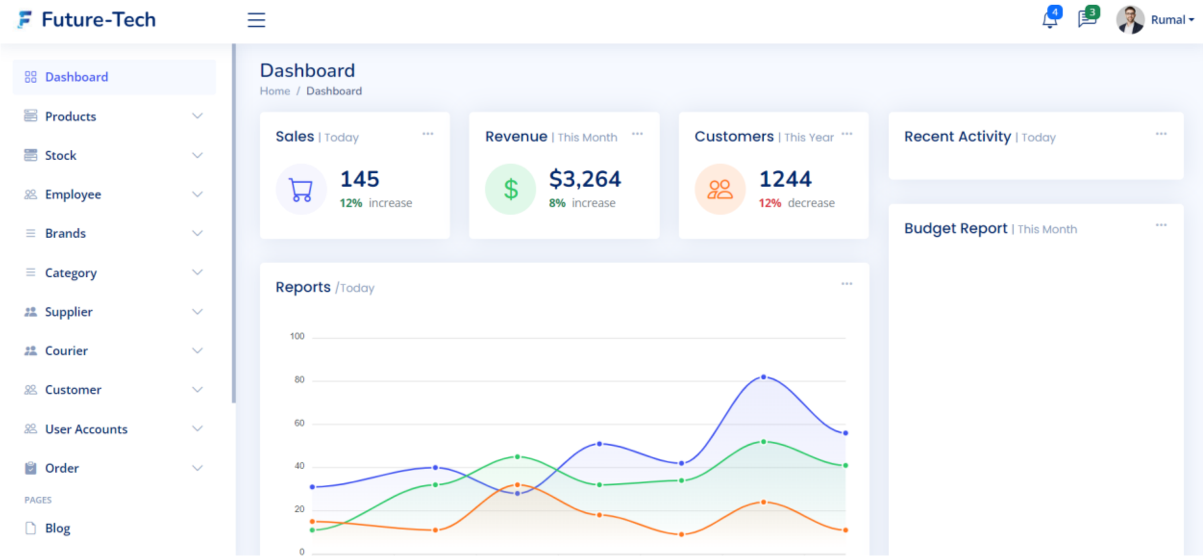
6.6.5 Checkout page



Products	Total
MSI GE76 RAIDER 11UE-1056 Laptop	Rs:485,000.00
24" iMac with Retina 4.5K display	Rs:343,000.00
Total	Rs:828,000

The Checkout Page is a critical component of the online store's user interface, serving as the final step in the purchasing process. Once users have reviewed and confirmed the items in their shopping cart, they are directed to the Checkout Page to provide the necessary information for order fulfillment. The Checkout Page aims to make the payment and order completion process smooth and secure.

6.6.5 Management System home page

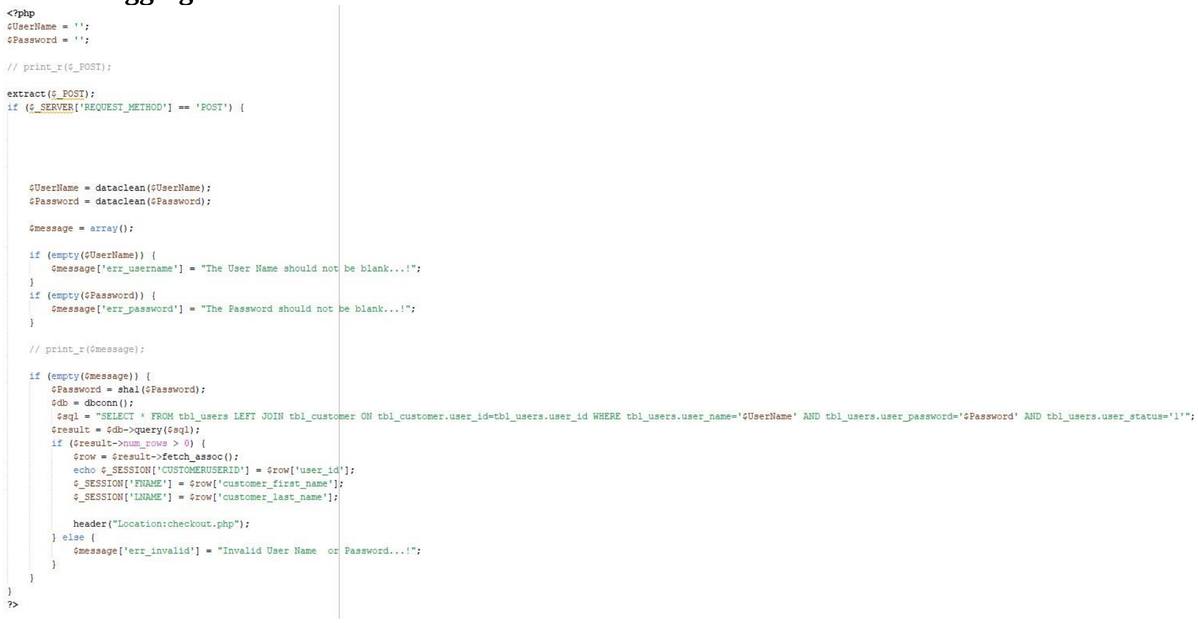


The Management System Home Page serves as the central hub for administrators and authorized personnel to access and manage various aspects of the online store or business. It provides a comprehensive overview of the system's functionalities and quick access to key management tasks. The design of the Management System Home Page is focused on efficiency, usability, and easy navigation.

The design of the Management System Home Page ensures that administrators can efficiently monitor and control various aspects of the business. It provides a user-friendly interface that streamlines complex management tasks, ultimately enhancing productivity and decision-making for the online store or business. The page's layout is responsive, ensuring a seamless experience across different devices and screen sizes. Overall, the Management System Home Page is a powerful tool for administrators, enabling them to maintain the smooth operation and success of the online store or business.

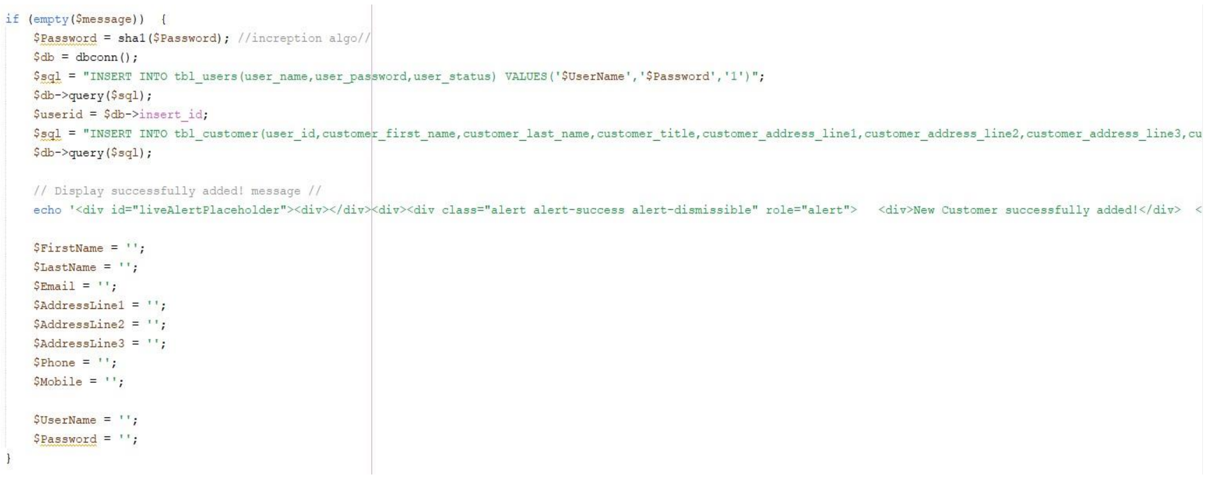
6.7 Backend Code Implementation

6.7.1 for logging



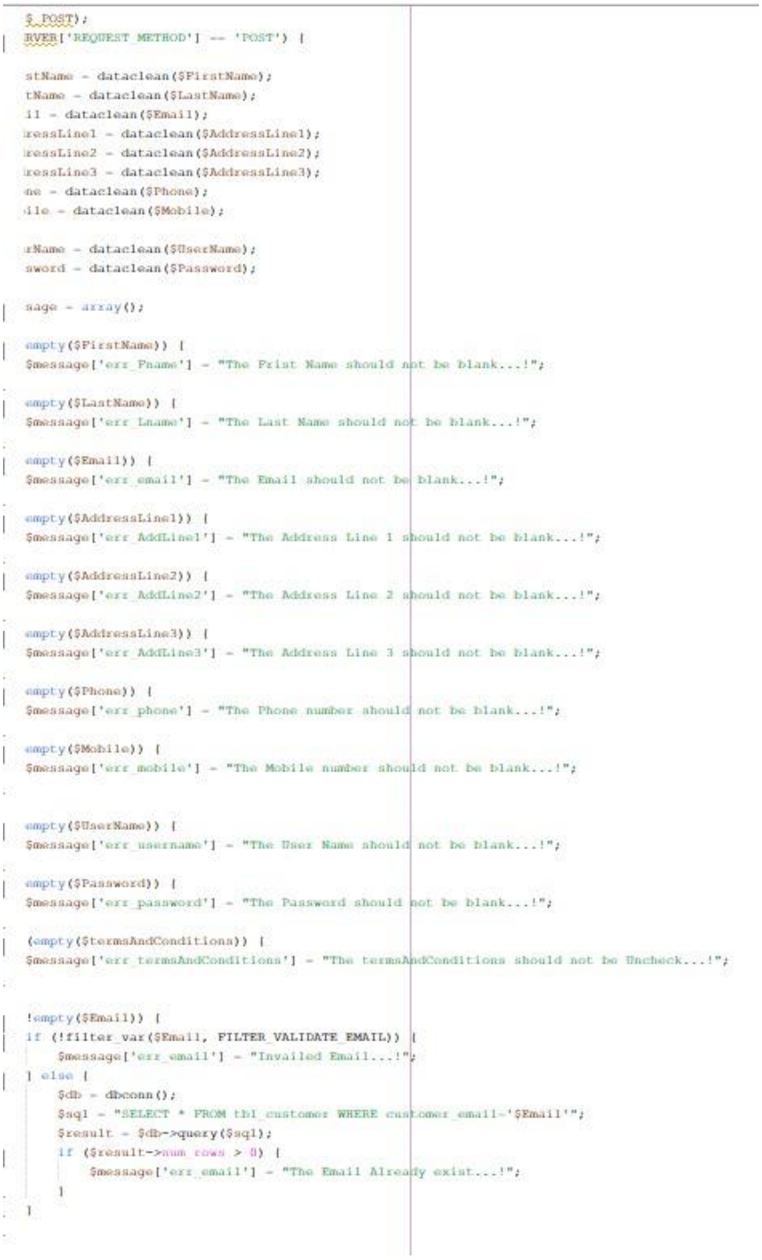
```
<?php
$UserName = '';
$Password = '';

// print_r($_POST);

extract($_POST);
if ($_SERVER['REQUEST_METHOD'] == 'POST') {




    $UserName = dataclean($UserName);
    $Password = dataclean($Password);

    $message = array();

    if (empty($UserName)) {
        $message['err_username'] = "The User Name should not be blank...!";
    }
    if (empty($Password)) {
        $message['err_password'] = "The Password should not be blank...!";
    }

    // print_r($message);

    if (empty($message)) {
        $Password = sha1($Password);
        $db = dbconn();
         $sql = "SELECT * FROM tbl_users LEFT JOIN tbl_customer ON tbl_customer.user_id=tbl_users.user_id WHERE tbl_users.user_name='$UserName' AND tbl_users.user_password='$Password' AND tbl_users.user_status='1'";
        $result = $db->query($sql);
        if ($result->num_rows > 0) {
            $row = $result->fetch_assoc();
            echo $_SESSION['CUSTOMERUSERID'] = $row['user_id'];
            $_SESSION['FNAME'] = $row['customer_first_name'];
            $_SESSION['LNAME'] = $row['customer_last_name'];

            header("Location:checkout.php");
        } else {
            $message['err_invalid'] = "Invalid User Name  or Password...!";
        }
    }
}
?>
```

6.7.2 for register



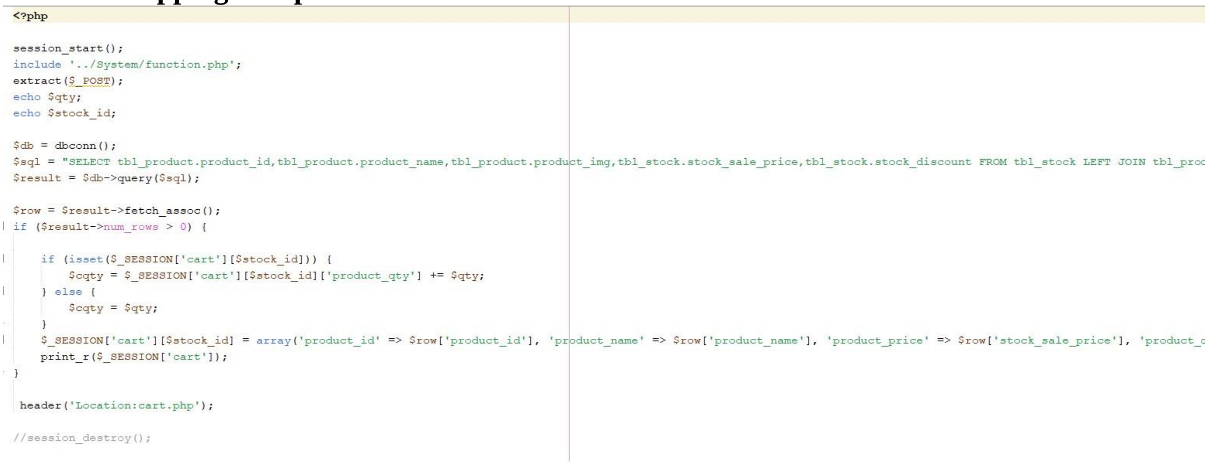
```
if (empty($message))  {
    $Password = sha1($Password); //inception algo//
    $db = dbconn();
    $sql = "INSERT INTO tbl_users(user_name,user_password,user_status) VALUES('$UserName','$Password','1')";
    $db->query($sql);
    $userid = $db->insert_id;
    $sql = "INSERT INTO tbl_customer(user_id,customer_first_name,customer_last_name,customer_title,customer_address_line1,customer_address_line2,customer_address_line3,cu
    $db->query($sql);

    // Display successfully added! message //
    echo '<div id="liveAlertPlaceholder"><div></div><div><div class="alert alert-success alert-dismissible" role="alert">   <div>New Customer successfully added!</div>  <

    $FirstName = '';
    $LastName = '';
    $Email = '';
    $AddressLine1 = '';
    $AddressLine2 = '';
    $AddressLine3 = '';
    $Phone = '';
    $Mobile = '';

    $UserName = '';
    $Password = '';
}
```

6.7.3 for data validation



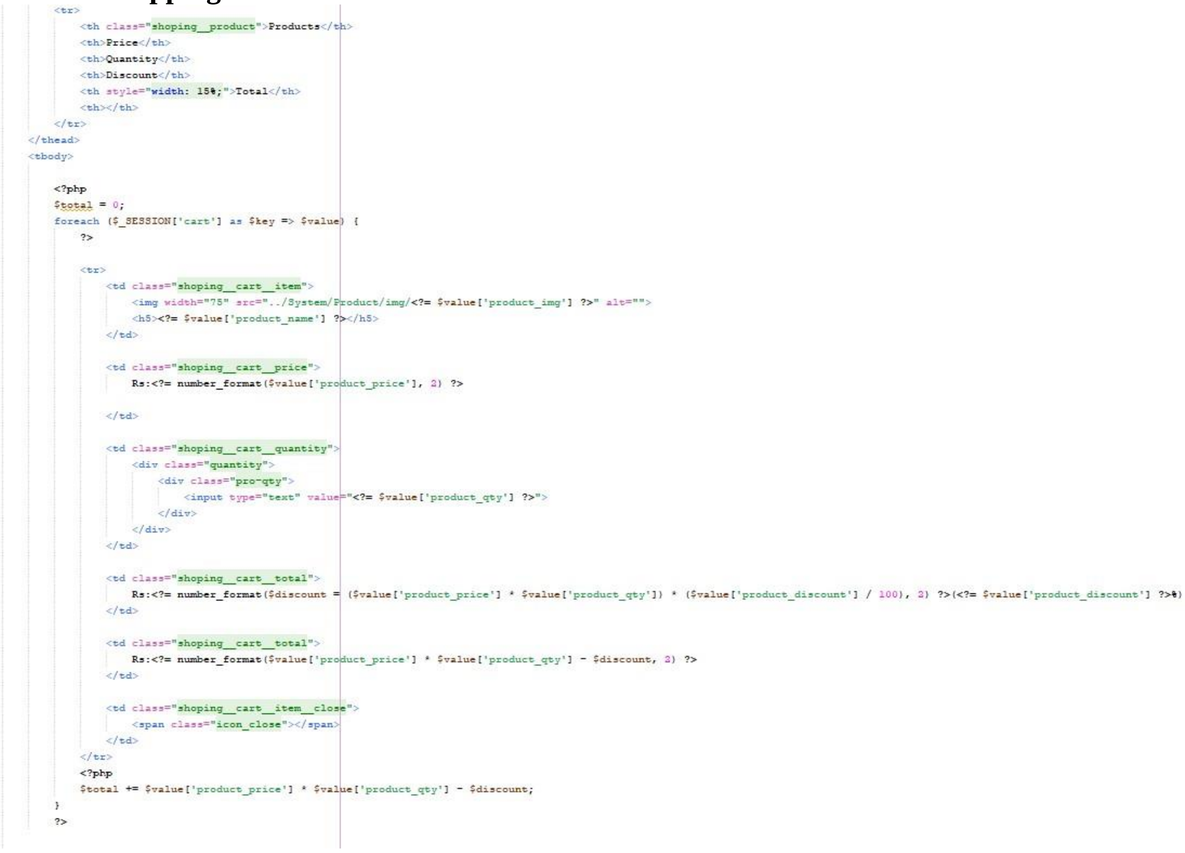
```
$_POST);
RVER['REQUEST_METHOD'] == 'POST') {

stName = dataclean($FirstName);
tName = dataclean($LastName);
il = dataclean($Email);
ressLine1 = dataclean($AddressLine1);
ressLine2 = dataclean($AddressLine2);
ressLine3 = dataclean($AddressLine3);
ne = dataclean($Phone);
ile = dataclean($Mobile);

rName = dataclean($UserName);
sword = dataclean($Password);

sage = array();

empty($FirstName)) {
$message['err_Fname'] = "The Frist Name should not be blank...!";

empty($LastName)) {
$message['err_Lname'] = "The Last Name should not be blank...!";

empty($Email)) {
$message['err_email'] = "The Email should not be blank...!";

empty($AddressLine1)) {
$message['err_AddLine1'] = "The Address Line 1 should not be blank...!";

empty($AddressLine2)) {
$message['err_AddLine2'] = "The Address Line 2 should not be blank...!";

empty($AddressLine3)) {
$message['err_AddLine3'] = "The Address Line 3 should not be blank...!";

empty($Phone)) {
$message['err_phone'] = "The Phone number should not be blank...!";

empty($Mobile)) {
$message['err_mobile'] = "The Mobile number should not be blank...!";


empty($UserName)) {
$message['err_username'] = "The User Name should not be blank...!";

empty($Password)) {
$message['err_password'] = "The Password should not be blank...!";

(empty($termsAndConditions)) {
$message['err_termsAndConditions'] = "The termsAndConditions should not be Uncheck...!";


!empty($Email)) {
if (!filter_var($Email, FILTER_VALIDATE_EMAIL)) {
    $message['err_email'] = "Invailed Email...!";
} else {
    $db = dbconn();
    $sql = "SELECT * FROM tbl_customer WHERE customer_email='$Email'";
    $result = $db->query($sql);
    if ($result->num_rows > 0) {
        $message['err_email'] = "The Email Already exist...!";
    }
}
```

6.7.4 for shopping cart process



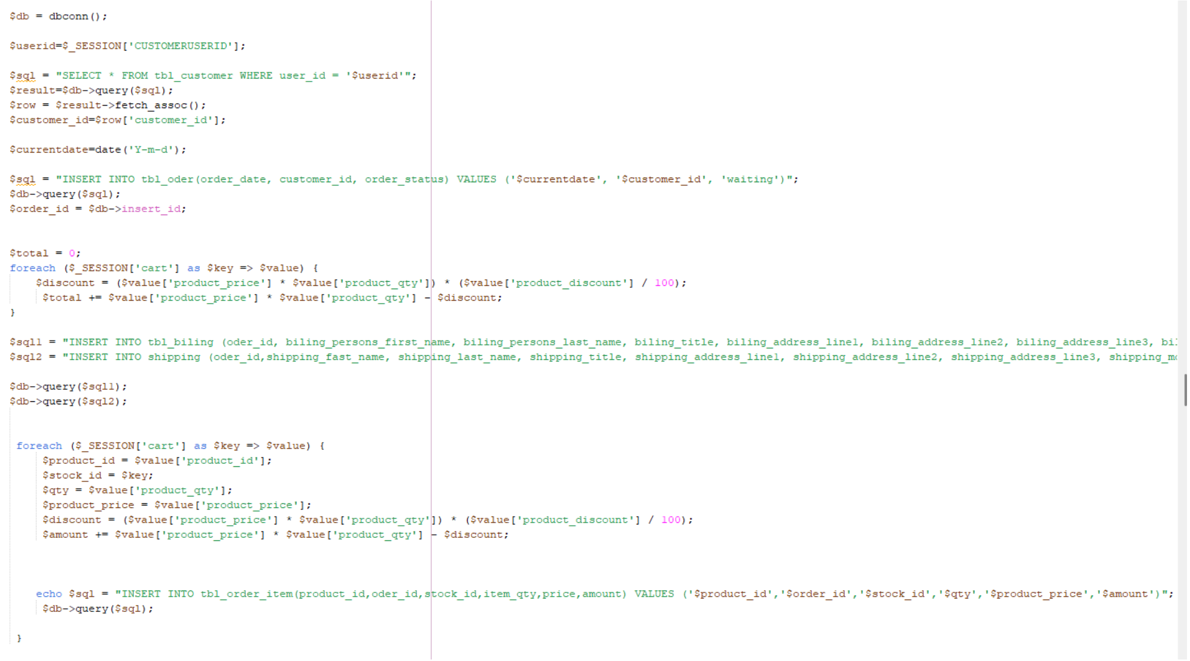
```php
<?php

session_start();
include '../System/function.php';
extract($_POST);
echo $qty;
echo $stock_id;

$db = dbconn();
$sql = "SELECT tbl_product.product_id,tbl_product.product_name,tbl_product.product_img,tbl_stock.stock_sale_price,tbl_stock.stock_discount FROM tbl_stock LEFT JOIN tbl_proc
$result = $db->query($sql);

$row = $result->fetch_assoc();
if ($result->num_rows > 0) {

    if (isset($_SESSION['cart'][$stock_id])) {
        $cqty = $_SESSION['cart'][$stock_id]['product_qty'] += $qty;
    } else {
        $cqty = $qty;
    }
    $_SESSION['cart'][$stock_id] = array('product_id' => $row['product_id'], 'product_name' => $row['product_name'], 'product_price' => $row['stock_sale_price'], 'product_c
    print_r($_SESSION['cart']);
}

 header('Location:cart.php');

//session_destroy();
```

6.7.5 for shopping cart



```php
    <tr>
        <th class="shoping__product">Products</th>
        <th>Price</th>
        <th>Quantity</th>
        <th>Discount</th>
        <th style="width: 15%;">Total</th>
        <th></th>
    </tr>
</thead>
<tbody>

    <?php
    $total = 0;
    foreach ($_SESSION['cart'] as $key => $value) {
        ?>

        <tr>
            <td class="shoping__cart__item">
                <img width="75" src="../System/Product/img/<?= $value['product_img'] ?>" alt="">
                <h5><?= $value['product_name'] ?></h5>
            </td>

            <td class="shoping__cart__price">
                Rs:<?= number_format($value['product_price'], 2) ?>

            </td>

            <td class="shoping__cart__quantity">
                <div class="quantity">
                    <div class="pro-qty">
                        <input type="text" value="<?= $value['product_qty'] ?>">
                    </div>
                </div>
            </td>

            <td class="shoping__cart__total">
                Rs:<?= number_format($discount = ($value['product_price'] * $value['product_qty']) * ($value['product_discount'] / 100), 2) ?>(<?= $value['product_discount'] ?>%)
            </td>

            <td class="shoping__cart__total">
                Rs:<?= number_format($value['product_price'] * $value['product_qty'] - $discount, 2) ?>
            </td>

            <td class="shoping__cart__item__close">
                <span class="icon_close"></span>
            </td>
        </tr>
        <?php
        $total += $value['product_price'] * $value['product_qty'] - $discount;
    }
    ?>
```

6.7.5 for checkout



```
$db = dbconn();

$userid=$_SESSION['CUSTOMERUSERID'];

$sql = "SELECT * FROM tbl_customer WHERE user_id = '$userid'";
$result=$db->query($sql);
$row = $result->fetch_assoc();
$customer_id=$row['customer_id'];

$currentdate=date('Y-m-d');

$sql = "INSERT INTO tbl_oder(order_date, customer_id, order_status) VALUES ('$currentdate', '$customer_id', 'waiting')";
$db->query($sql);
$order_id = $db->insert_id;


$total = 0;
foreach ($_SESSION['cart'] as $key => $value) {
    $discount = ($value['product_price'] * $value['product_qty']) * ($value['product_discount'] / 100);
     $total += $value['product_price'] * $value['product_qty'] - $discount;
}

$sql1 = "INSERT INTO tbl_biling (oder_id, biling_persons_first_name, biling_persons_last_name, biling_title, biling_address_line1, biling_address_line2, biling_address_line3, bil
$sql2 = "INSERT INTO shipping (oder_id,shipping_fast_name, shipping_last_name, shipping_title, shipping_address_line1, shipping_address_line2, shipping_address_line3, shipping_m

$db->query($sql1);
$db->query($sql2);


 foreach ($_SESSION['cart'] as $key => $value) {
     $product_id = $value['product_id'];
     $stock_id = $key;
     $qty = $value['product_qty'];
     $product_price = $value['product_price'];
     $discount = ($value['product_price'] * $value['product_qty']) * ($value['product_discount'] / 100);
     $amount += $value['product_price'] * $value['product_qty'] - $discount;



    echo $sql = "INSERT INTO tbl_order_item(product_id,oder_id,stock_id,item_qty,price,amount) VALUES ('$product_id','$order_id','$stock_id','$qty','$product_price','$amount')";
     $db->query($sql);

 }
```

Chapter 7 -Testing and Verification

7.1 Introduction

The testing and verification phase is a critical stage in the development of the shopping cart and stock management web application. This chapter provides an overview of the testing process, aiming to ensure the functionality, reliability, and quality of the application.

7.2 Aim of Testing

The primary aim of testing is to verify that the web application meets the defined requirements and functions as intended. The testing process focuses on identifying and resolving any issues or defects to ensure a high-quality and reliable system. Key objectives of testing include:

Functionality Testing: Verify that all functionalities of the web application work as expected and meet the specified requirements. This includes testing features such as user registration, product search, shopping cart functionality, payment processing, order tracking, and inventory management.

User Interface Testing: Ensure that the user interface is intuitive, user-friendly, and responsive across different devices and screen sizes. Test the layout, navigation, and interactions to provide a seamless and enjoyable user experience.

Performance Testing: Evaluate the performance of the web application under different load conditions to ensure it can handle the expected user traffic. Measure response times, throughput, and resource utilization to identify potential bottlenecks and optimize performance.

Security Testing: Assess the security measures implemented in the web application to identify vulnerabilities or weaknesses. Conduct tests to ensure secure user authentication, data encryption, protection against common web application attacks (e.g., SQL injection, cross-site scripting), and compliance with relevant security standards.

Compatibility Testing: Test the web application across different web browsers, operating systems, and devices to ensure compatibility and consistent functionality. Verify that the application functions correctly and displays properly across various platforms.

Integration Testing: Verify the integration of external systems, such as payment gateways and shipping providers, to ensure seamless communication and data exchange. Test end-to-end scenarios to validate the integration points and ensure smooth workflow across the application and external systems.

Error Handling and Exception Testing: Assess the web application's ability to handle errors, exceptions, and edge cases gracefully. Verify that appropriate error messages are displayed, and the system recovers gracefully from failures.

User Acceptance Testing (UAT): Involve end-users or representative stakeholders in testing the application to gather feedback and ensure it meets their expectations. Obtain their validation of the functionality, usability, and overall satisfaction with the web application.

By conducting thorough testing based on these objectives, the developer can identify and address any issues, ensuring a high-quality and reliable shopping cart and stock management web application. Testing is crucial for delivering a robust and user-friendly system that meets the needs of Future Tech and its customers.

7.3 Test Plan

The test plan outlines the approach, scope, and objectives of the testing phase for the shopping cart and stock management web application. It provides a structured framework for conducting various tests to validate the application's functionality, performance, security, and user experience. Here is an overview of the key components of the test plan:

1. Test Objectives:
   - Define the specific goals and objectives of the testing phase, such as validating functionality, ensuring performance, assessing security measures, and verifying user experience.

2. Test Scope:
   - Identify the components, functionalities, and modules of the web application that will be included in the testing.
   - Determine the boundaries and limitations of the testing, specifying any excluded features or components.

3. Test Approach:
   - Determine the testing approach, such as black-box testing, white-box testing, or a combination of both.
   - Define the testing techniques and methods to be employed, such as functional testing, usability testing, integration testing, and performance testing.

4. Test Environment:
   - Specify the hardware, software, and network environment required for conducting the tests.
   - Identify any test tools or frameworks that will be utilized to support the testing activities.

5. Test Scenarios:
   - Define the specific test scenarios that will be executed, covering different aspects of the web application's functionality, user interactions, and system behavior.
   - Document the expected inputs, expected outputs, and the steps to be performed for each test scenario.

6. Test Data:
   - Identify the test data required to execute the test scenarios effectively.
   - Specify the sources of test data and any modifications or transformations required to ensure test data integrity.

7. Test Execution Schedule:
   - Define the timeline and sequence for executing the test scenarios.
   - Allocate resources and determine the responsibilities of individuals or teams involved in the testing process.

8. Defect Management:

- Establish a process for reporting, tracking, and resolving defects identified during testing.
- Define the severity and priority levels for different types of defects.
- Determine the criteria for determining when a defect is considered resolved.

9. Test Reporting:

- Specify the format and contents of test reports that will be generated during the testing phase.
- Define the key metrics and measurements that will be captured to assess the progress and results of the testing.

10. Risks and Mitigation:

- Identify potential risks and challenges that may impact the testing process or the quality of the web application.
- Define mitigation strategies and contingency plans to address these risks and minimize their impact.

The test plan serves as a roadmap for the testing activities, ensuring that comprehensive and systematic testing is conducted to validate the shopping cart and stock management web application. It provides a structured approach to identify and address any issues, ensuring a high-quality and reliable system for Future Tech and its customers.

7.4 Validation Test Plan

7.4.1 Test Plan of the Login Form

1. Test Scope:

- This test plan is designed to verify the functionality and security of the login form.
- The focus is on validating the login process, handling various scenarios, and ensuring data integrity.

2. Test Objectives:

- Verify that users can successfully log in with valid credentials.
- Validate the handling of invalid login attempts.
- Ensure appropriate error messages are displayed for invalid inputs.
- Test the security measures implemented to protect against unauthorized access.
- Verify that the login form functions correctly across different devices and browsers.

3. Test Environment:

- Operating System: Windows 11
- Browser: chrome
- Device: Laptop

4. Test Scenarios:

Scenario 1: Successful Login

- Description: Test the login process with valid credentials.
- Test Steps:

  1. Enter valid username and password.
  2. Click on the "Login" button.
  3. Verify that the user is successfully logged in and redirected to the appropriate page.

Scenario 2: Invalid Username or Password

- Description: Test the handling of invalid username or password inputs.
- Test Steps:

  1. Enter an invalid username or password.
  2. Click on the "Login" button.
  3. Verify that the appropriate error message is displayed.
  4. Verify that the user is not logged in and remains on the login page.

Scenario 3: Account Lockout

- Description: Test the account lockout functionality by the admin.
- Test Steps:

  1. Enter a valid username or password multiple times.
  2. Verify that the account is locked.

Scenario 4: Cross-browser and Cross-device Compatibility

- Description: Test the login form on different browsers and devices to ensure compatibility.
- Test Steps:

1. Access the login form using different browsers (e.g., Chrome, Firefox, Safari, Edge).
2. Test the login form on different devices (e.g., desktop, tablet, mobile).
3. Verify that the login form functions correctly and displays properly on each browser and device.

5. Test Data:

- Prepare test data for valid and invalid inputs, including usernames, passwords, and email addresses.

6. Test Execution:

- Execute the test scenarios described above, documenting the actual results and any issues encountered.
- Capture screenshots or videos if necessary to provide visual evidence of the test execution.

7. Test Reporting:

- Compile the test results, including the executed test scenarios, actual results, and any defects or observations.
- Provide detailed documentation on any issues encountered during testing.
- Report any security vulnerabilities or concerns identified during testing.

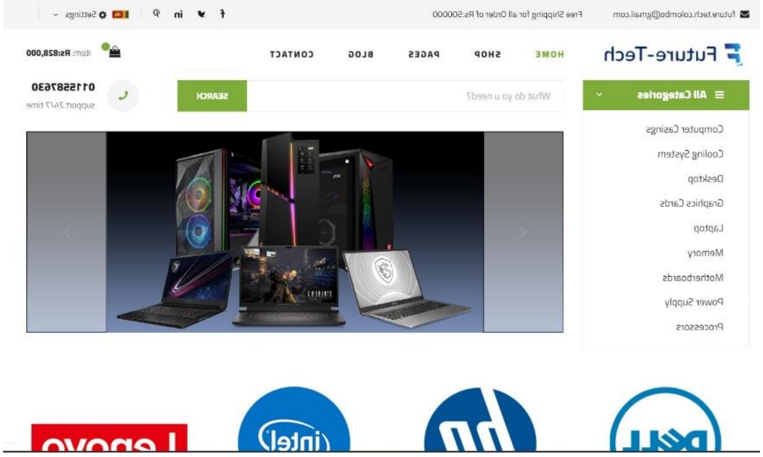
8. Test Completion Criteria:

- All test scenarios have been executed and documented.
- The actual results match the expected results for each scenario.
- Any identified defects or issues have been reported and resolved.

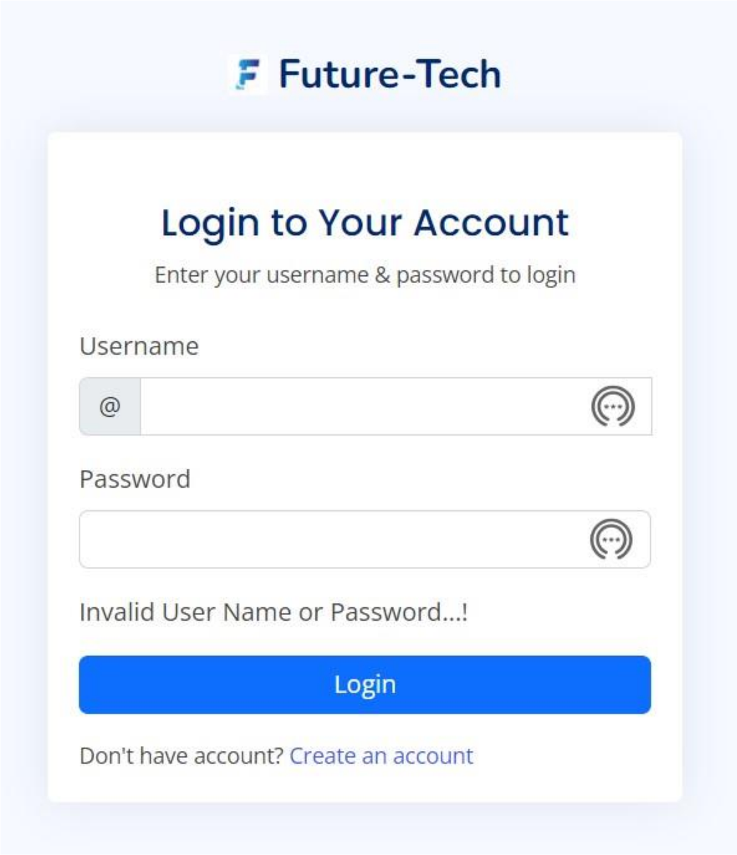
7.5 Validation Test Case

7.5.1 Test Case of Login

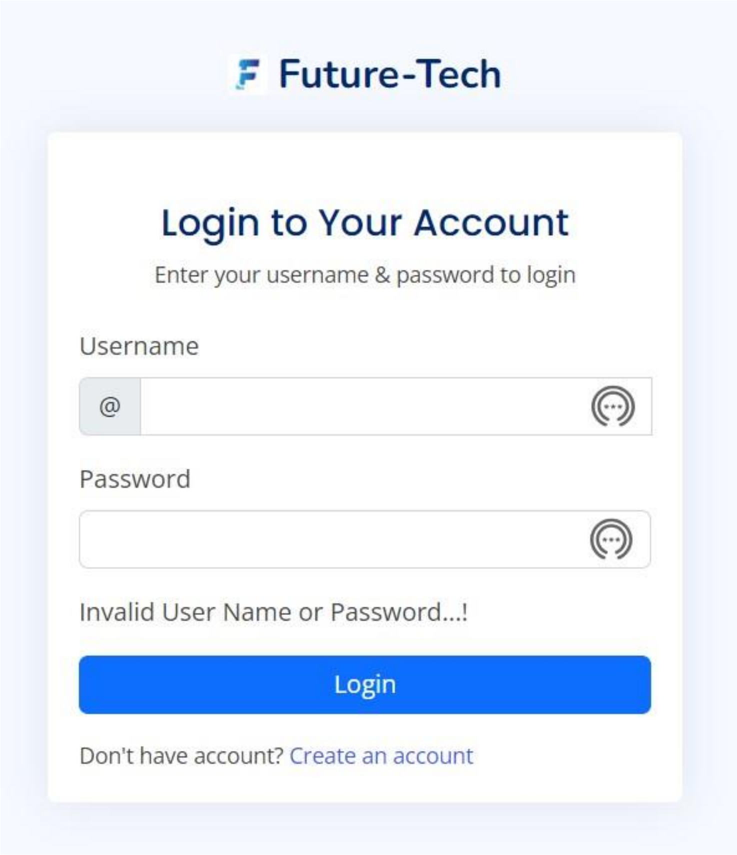
Format Followed:

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC1	Login with **Valid Credentials**	The user is successfully logged in. redirected to the home page. Able to view the home page.
Actual Output	 future.tech.colombo@gmail.com Free Shipping for all Order of Rs.500000 Settings Future-Tech HOME SHOP PAGES BLOG CONTACT item: Rs:8828,000 All Categories What do u need? SEARCH 0115587630 support 24/7 time Computer Casings Cooling System Desktop Graphics Cards Laptop Memory Motherboards Power Supply Processors Validation has been passed.	

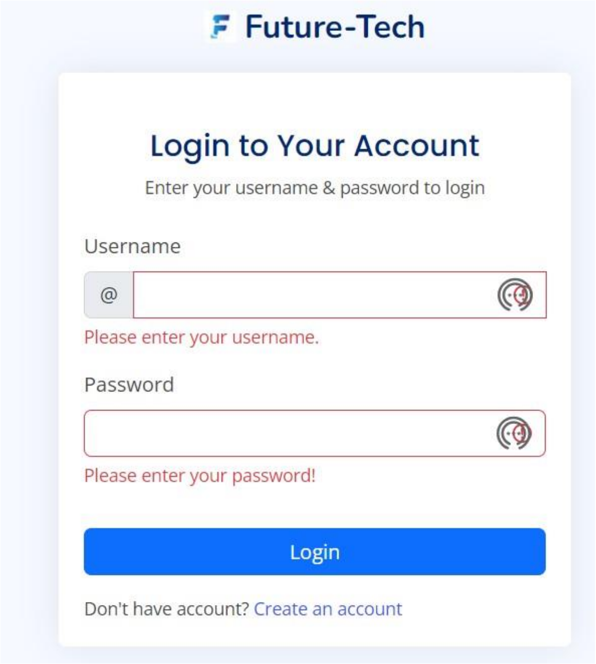
Format Followed:

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm Date:18/07/2023
Test No:	Description	Expected Output
TC2	Verify the login functionality with an **invalid Password** and valid user name.	- After clicking the "Login" button, the system should validate the entered credentials. - If the credentials are invalid, an error message should be displayed indicating the login failure. - The user should remain on the login page without being logged in. - The user should have the option to retry entering valid credentials or reset the password.
Actual Output	 Future-Tech Login to Your Account Enter your username & password to login Username @ Password Invalid User Name or Password...! Login Don't have account? Create an account Validation has been passed	

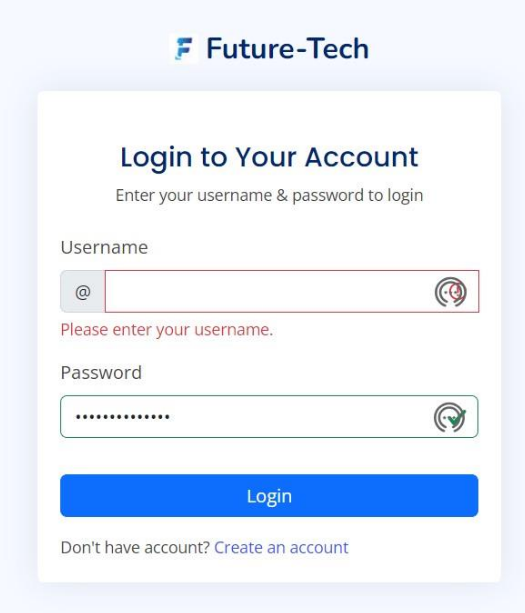
Format Followed:

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC3	Verify the login functionality with an **invalid User name** and valid Password.	• After clicking the "Login" button, the system should validate the entered credentials. • If the credentials are invalid, an error message should be displayed indicating the login failure. • The user should remain on the login page without being logged in. • The user should have the option to retry entering valid credentials or reset the password.
Actual Output	 Future-Tech Login to Your Account Enter your username & password to login Username @ Password Invalid User Name or Password...! Login Don't have account? Create an account Validation has been passed	

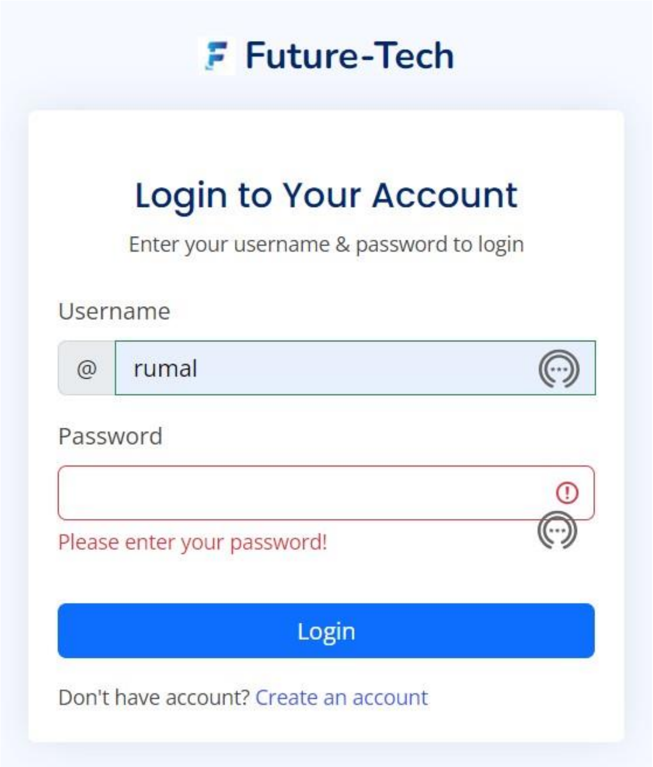
Format Followed:

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC4	Try to log in with **fields empty**	• The system should validate the login form fields and display appropriate error messages for each empty field. • The user should remain on the login page without being logged in.
Actual Output	 Future-Tech Login to Your Account Enter your username & password to login Username @ Please enter your username. Password Please enter your password! Login Don't have account? Create an account Validation has been passed	

Format Followed:

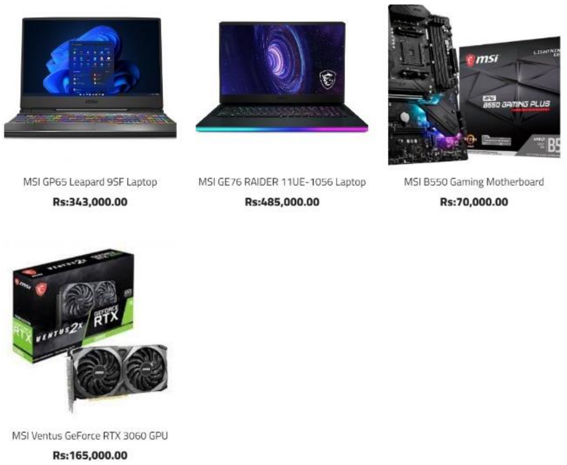
Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC5	Log in **without a username** and valid password.	• The system should validate the login form fields and display appropriate error messages for each empty field. • The user should remain on the login page without being logged in.
Actual Output	 Future-Tech Login to Your Account Enter your username & password to login Username @ Please enter your username. Password •••••••••••••• Login Don't have account? Create an account Validation has been passed	

Format Followed:

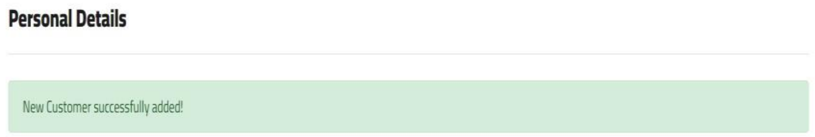
Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC6	Login **without a password** and a valid username.	- The system should validate the login form fields and display appropriate error messages for each empty field. - The user should remain on the login page without being logged in.
Actual Output	 Future-Tech Login to Your Account Enter your username & password to login Username @ rumal Password Please enter your password! Login Don't have account? Create an account Validation has been passed	

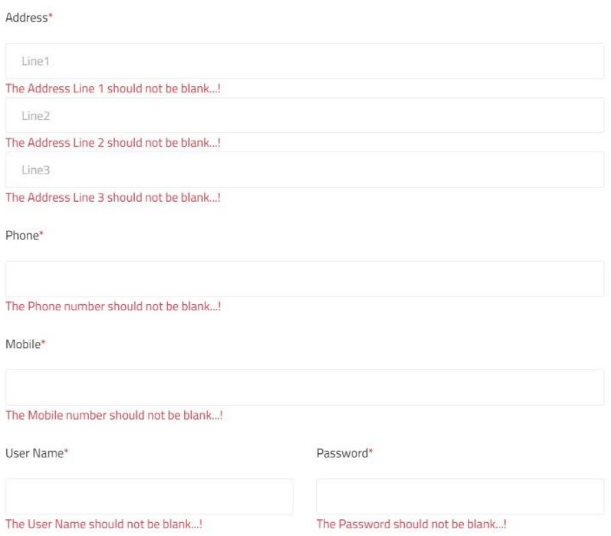
7.5.2 Test Case of Main Menu

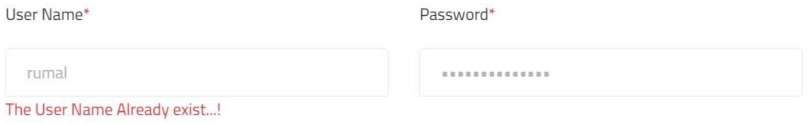
Format Followed:

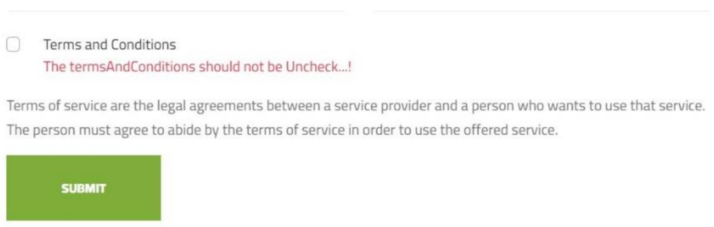
Executed By: Rumal Tharinda Nayanatha		Time:9.13pm Date:18/07/2023
Test No:	Description	Expected Output
TC7	Verify the main menu navigation functionality, Click on the MSI brand name.	• For each main menu item, the user should be directed to the corresponding page. • The content of the page should match the selected menu item. • The active menu item should be visually highlighted or indicated. • The user should be able to navigate back to the home page by clicking on the logo or home menu item.
Actual Output	 MSI GP65 Leapard 9SF Laptop Rs:343,000.00 MSI GE76 RAIDER 11UE-1056 Laptop Rs:485,000.00 MSI B550 Gaming Motherboard Rs:70,000.00 MSI Ventus GeForce RTX 3060 GPU Rs:165,000.00 Displaying MSI product only therefore Validation has been passed.	

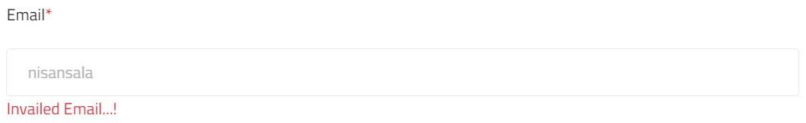
7.5.3 Test Case of Customer registration

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC8	Verify the customer registration functionality by clicking the register button with valid input fields.	• After clicking the "Register" button, the system should validate the customer details. • If the details are valid, the customer should be successfully registered. • The customer should receive a confirmation message. • The system should redirect the customer to the login page.
Actual Output	 Personal Details New Customer successfully added! Validation has been passed	

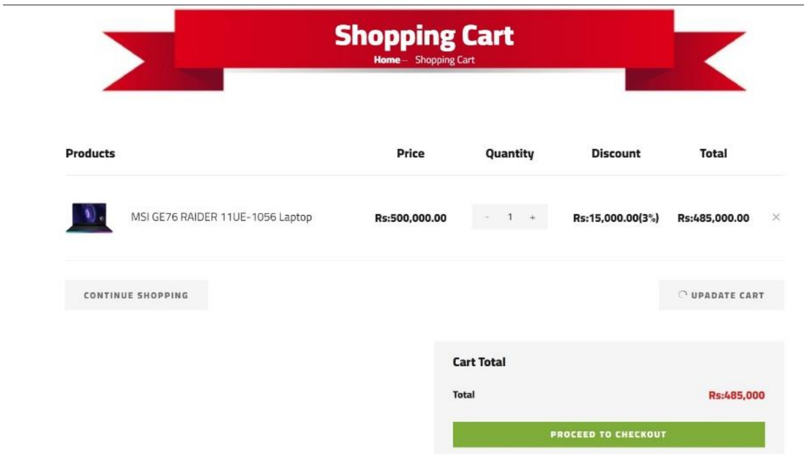
Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC9	Verify the customer registration functionality by clicking the register button with **empty fields.**	- System should display appropriate error messages for each empty field.
Actual Output	 Address* Line1 The Address Line 1 should not be blank...! Line2 The Address Line 2 should not be blank...! Line3 The Address Line 3 should not be blank...! Phone* The Phone number should not be blank...! Mobile* The Mobile number should not be blank...! User Name* The User Name should not be blank...! Password* The Password should not be blank...! Validation has been passed	

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC10	Verify the customer registration functionality by entering the **existing user name.**	- After clicking the "Register" button, the system should validate the customer details. - The system should display appropriate error messages. - The customer should remain on the registration page without being logged in. -
Actual Output	 User Name* rumal The User Name Already exist...! Password* •••••••••••••• Validation has been passed	

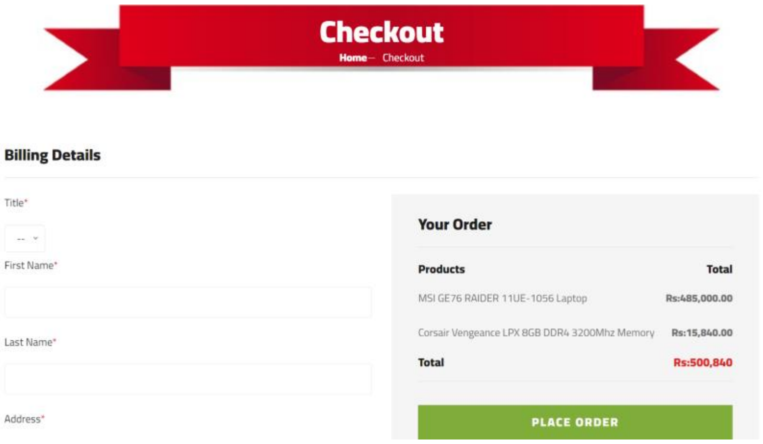
Executed By: Rumal Tharinda Nayanatha		Time:9.13pm Date:18/07/2023
Test No:	Description	Expected Output
TC11	Verify the customer registration functionality by clicking the register button **without agreeing to the terms and conditions**.	• After clicking the "Register" button, the system should validate the customer details. • The system should display appropriate error messages. • The customer should remain on the registration page without being logged in.
Actual Output	 □ Terms and Conditions The termsAndConditions should not be Uncheck...! Terms of service are the legal agreements between a service provider and a person who wants to use that service. The person must agree to abide by the terms of service in order to use the offered service. Validation has been passed	

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm Date:18/07/2023
Test No:	Description	Expected Output
TC12	Verify the customer registration functionality by entering the **invalid email.**	• After clicking the "Register" button, the system should validate the customer details. • The system should display appropriate error messages. • The customer should remain on the registration page without being logged in. •
Actual Output	 Email* nisansala Invailed Email...! Validation has been passed	

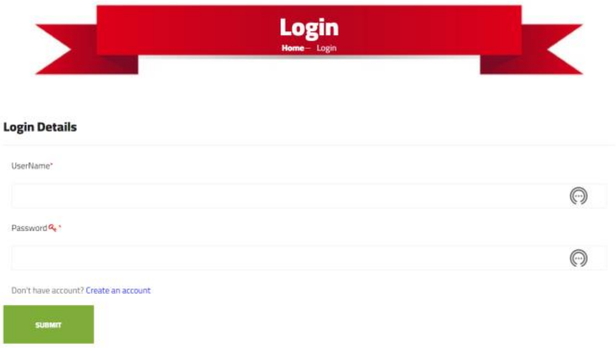
7.5.4 Test Case of adding items to cart

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC13	Test the functionality of adding items to the cart in the online store.	• When the "Add to Cart" button is clicked for a product, the system should add the selected item to the user's cart. • The cart should display the added item with its details, such as name, price, and quantity. • The total price of the cart should be updated to reflect the cost of the added item.
Actual Output	 Shopping Cart Home– Shopping Cart Products Price Quantity Discount Total MSI GE76 RAIDER 11UE-1056 Laptop Rs:500,000.00 - 1 + Rs:15,000.00(3%) Rs:485,000.00 × CONTINUE SHOPPING UPADATE CART Cart Total Total Rs:485,000 PROCEED TO CHECKOUT Validation has been passed	

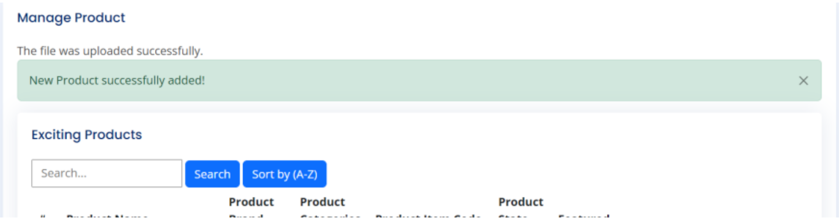
7.5.5 Test Case of checkout process

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC14	Test the checkout process in the online store to ensure a smooth and successful.	- When the "Add to Cart" button is clicked for a product, the system should add the selected item to the user's cart. - The cart should display the added item with its details, such as name, price, and quantity. - The total price of the cart should be updated to reflect the cost of the added item.
Actual Output	 Checkout Home— Checkout Billing Details Title* -- First Name* Last Name* Address* Your Order Products Total MSI GE76 RAIDER 11UE-1056 Laptop Rs:485,000.00 Corsair Vengeance LPX 8GB DDR4 3200Mhz Memory Rs:15,840.00 Total Rs:500,840 PLACE ORDER Validation has been passed	

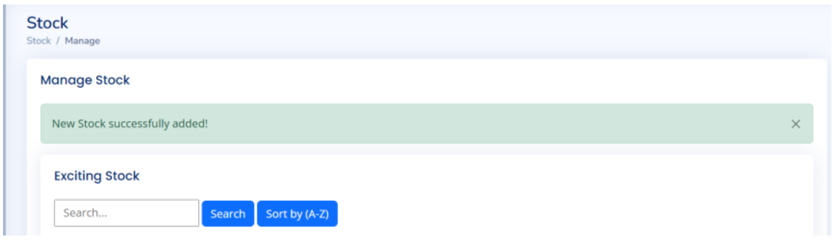
7.5.6 Test Case of logout

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC15	Test the logout functionality in the online store to verify if the user can successfully log out of their account.	• When the user clicks on the "Logout" button the system should log them out of their account. • After logout, the user should be redirected to the login page or the home page. • Upon successful logout, the user should not be able to access protected pages or perform actions that require authentication until they log in again.
Actual Output	 Login Home – Login Login Details UserName* Password * Don't have account? Create an account SUBMIT After clicking logout button, the system logout and redirect to login page therefore Validation has been passed.	

7.5.7 Test Case of adding new product

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm
		Date:18/07/2023
Test No:	Description	Expected Output
TC16	Test the functionality of adding a new product to the online store's inventory.	• When the user with appropriate privileges clicks on the accesses the "Add Product" page, they should be presented with a form to enter product details. • After filling in all the required product information (e.g., name, description, price, category, etc.), the user should be able to submit the form. • Upon successful submission, the new product should be added to the product inventory database. • The user should receive a confirmation message indicating that the product addition was successful.
Actual Output	 Manage Product The file was uploaded successfully. New Product successfully added! Exciting Products Search... Search Sort by (A-Z) Product Product Product Validation has been passed	

7.6.8 Test Case of adding new stocks

Executed By: Rumal Tharinda Nayanatha		Time:9.13pm Date:18/07/2023
Test No:	Description	Expected Output
TC17	Test the functionality of adding new stocks to an existing product in the online	• When the user with appropriate privileges clicks on the accesses the "Add Stocks" page, they should be presented with a form to select the product and enter the quantity of stocks to be added. • The user should be able to select the desired product from a dropdown list or search bar, which displays the existing products in the inventory. • After selecting the product and entering the quantity of stocks, the user should be able to submit the form. • Upon successful submission, the new stocks should be added to the selected product's stock count in the inventory database. • The user should receive a confirmation message indicating that the stocks addition was successful., category, etc.), the user should be able to submit the form. • Upon successful submission, the new product should be added to the product inventory database. • The user should receive a confirmation message indicating that the product addition was successful.
Actual Output	 Stock Stock / Manage Manage Stock New Stock successfully added! × Exciting Stock Search... Search Sort by (A-Z) Validation has been passed	

Chapter 8 -Evaluation and Conclusion

8.1 Summarized Evaluation

Throughout the development process of the shopping cart and stock management web application, an evaluation was conducted to assess its effectiveness and alignment with the objectives of Future Tech. Here is a summarized evaluation of the implemented solution:

Feasibility: The solution was found to be technically and operationally feasible, leveraging appropriate technologies and accommodating the business processes of Future Tech.

Functionality: The implemented web application successfully fulfilled the defined requirements, including user registration, product catalog, shopping cart functionality, secure payment integration, order processing, inventory management, customer relationship management, reporting, and mobile responsiveness.

User Experience: The user interface design provided an intuitive and seamless shopping experience, allowing customers to easily navigate through the website, search for products, add items to the cart, and proceed with secure transactions. The application's responsiveness and compatibility with different devices enhanced user satisfaction.

Integration: The solution effectively integrated with external systems and services, such as payment gateways and shipping providers, enabling smooth order processing and fulfillment.

Security: Appropriate security measures were implemented, including data encryption, secure user authentication, and compliance with data protection regulations, to safeguard customer data and protect against potential threats.

Performance and Scalability: The application demonstrated satisfactory performance and scalability, handling the expected user load and managing the growing product catalog efficiently. Caching mechanisms, query optimization, and load balancing techniques contributed to its overall responsiveness.

Maintenance: A maintenance plan was established to ensure ongoing support, monitoring, and enhancement of the web application. Regular backups, error handling mechanisms, and logging facilities were implemented to address potential issues and facilitate troubleshooting.

8.2 Conclusion

In conclusion, the development of the shopping cart and stock management web application has successfully addressed the challenges faced by Future Tech in the retail industry. The application's implementation followed a systematic approach, incorporating the waterfall methodology. Through careful planning, analysis, design, implementation, testing, and maintenance, Future Tech has achieved its objectives of enhancing sales, expanding the customer base, and improving operational efficiency.

The implemented solution provides Future Tech with a comprehensive online platform that offers a user-friendly interface, seamless shopping experience, secure transactions, and efficient stock management capabilities. The integration with external systems and services further enhances the application's functionality and customer satisfaction.

By utilizing tools and technologies such as HTML, CSS, PHP, NetBeans, SQL, phpMyAdmin, and XAMPP, Future Tech has created a robust and scalable web application. The database implementation ensures efficient data storage, retrieval, and manipulation, supporting the application's core functionalities.

The evaluation of the implemented solution has confirmed its alignment with the requirements and goals of Future Tech, validating its success in meeting customer expectations, improving business processes, and positioning Future Tech as a leading player in the computer parts retail industry.

Overall, the shopping cart and stock management web application provides Future Tech with a competitive edge in the evolving retail landscape. By continuously adapting and refining the application based on user feedback and market trends, Future Tech can sustain its growth, provide exceptional customer experiences, and solidify its position as a trusted and customer-centric online retailer.

9. References

References

- Intuit Limited., 2022. *Intuit.* [Online]
  Available at: https://quickbooks.intuit.com/global/oa/online-accounting-software-for-small-business/?cid=ppc_QBO_LK_Top_B_BMM_G&gclid=Cj0KCQjwk7ugBhDIARIsA
  [Accessed 21 06 2023].
- Leaf Group Ltd, 2023. *ehow.* [Online]
  Available at: https://www.ehow.com/about_541099%207_manual-vs_computerizedaccountingsytems.html
  [Accessed 05 07 2023].
- Adobe, 2019. *Adobe Commerce 2.4 Developer Guide.* [Online]
  Available at: https://devdocs.magento.com/
  [Accessed 1 02 2023].
- bootstrap, 2000. *bootstrap.* [Online]
  Available at: https://getbootstrap.com/docs/5.3/forms/validation/
  [Accessed 01 05 2023].
- codecademy, 2018. *catalog.* [Online]
  Available at: https://www.codecademy.com/catalog
  [Accessed 21 01 2023].
- codecalm, 2002. *codecalm.* [Online]
  Available at: https://tabler-icons.io/
  [Accessed 04 05 2023].
- fontawesome, 2017. *fontawesome.* [Online]
  Available at: https://fontawesome.com/v4/examples/#
  [Accessed 22 05 2023].
- Fuscaldo, D., Mar 31, 2023. *features-of-accounting-software.* [Online]
  Available at: https://www.business.com/articles/features-of-accounting-software/
  [Accessed 25 05 2023].
- investopedia, 2017. *feasibility-study.asp.* [Online]
  Available at: https://www.investopedia.com/terms/f/feasibility-study.asp
  [Accessed 09 06 2023].
- Newegg Inc, 2000. *Newegg.* [Online]
  Available at: https://www.newegg.com/
  [Accessed 29 04 2023].
- sense, 2015. *sense.* [Online]
  Available at: https://www.sense.lk/
  [Accessed 25 05 2023].
- shopify, 2021. *Shopify developer documentation.* [Online]
  Available at: https://shopify.dev/docs
  [Accessed 14 05 2023].
- thanish, 2014. *winsoft.* [Online]
  Available at: https://www.winsoft.lk/
  [Accessed 21 03 2023].

- university of kelaniya, 2002. *repository.* [Online]
  Available at: http://repository.kln.ac.lk/handle/123456789/16482
  [Accessed 08 04 2023].
- w3school, 2011. *w3school.* [Online]
  Available at: https://www.w3schools.com/php/default.asp
  [Accessed 14 04 2023].
- w3schools, 1999. *w3schools.* [Online]
  Available at: https://www.w3schools.com/html/
  [Accessed 23 05 2023].

- Li, X., & Huang, G. Q. (2017). E-commerce in the Construction Industry: A Review. Advanced Engineering Informatics, 33, 180-195.

- Chen, Y., et al. (2018). A Comparative Study of E-commerce Platforms: A Customer Experience Perspective. International Journal of Information Management, 43, 308-318.

- Smith, A., & Anderson, D. (2020). Enhancing User Experience in E-commerce Websites: A Literature Review. Journal of Computer Information Systems, 60(3), 246-254.

- Li, Y., et al. (2018). Factors Influencing Online Shopping Behavior: The Mediating Role of Purchase Intention. International Journal of Information Management, 39, 156-166.

10. Appendices

10.1 Interim Progress Reports

10.2 Company letter

10.3 Progress approval form and Project commencement meeting sheet.

10.3.1 Progress Approval Form

10.4 Supervisor Meetings

10.4.1 Project Commencement Meeting Sheet

Meeting	Criteria	Actions	Date
1.	Proposal Stage		
2.	Project Proposal Approval		
3.	Literature Review (Gathered resource documents)		

4.	Literature review (Report)		
5.	Trust Assessment		
6.	Interim Guidance		
7.	Interim Guidance 02		

8.	Final Guidance		